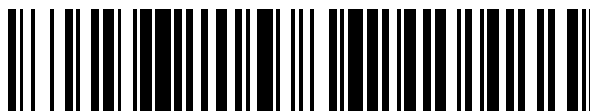


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 898 663**

51 Int. Cl.:

**H04N 19/30** (2014.01)

**H04N 19/70** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **19.06.2015 PCT/US2015/036614**

87 Fecha y número de publicación internacional: **23.12.2015 WO15196034**

96 Fecha de presentación y número de la solicitud europea: **19.06.2015 E 15741646 (2)**

97 Fecha y número de publicación de la concesión europea: **20.10.2021 EP 3158746**

54 Título: **Perfil, capa, nivel para la 0<sup>ésima</sup> capa de salida establecida en la codificación de vídeo**

30 Prioridad:

**20.06.2014 US 201462015210 P**  
**18.06.2015 US 201514743512**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**08.03.2022**

73 Titular/es:

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

72 Inventor/es:

**WANG, YE-KUI;**  
**RAMASUBRAMONIAN, ADARSH KRISHNAN y**  
**HENDRY, FNU**

74 Agente/Representante:

**ISERN JARA, Jorge**

ES 2 898 663 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

Perfil, capa, nivel para la 0<sup>ésima</sup> capa de salida establecida en la codificación de vídeo

### 5 Antecedentes

Campo tecnológico

10 La presente solicitud se relaciona con el campo de la codificación y compresión de video, por ejemplo, en codificadores y decodificadores de video. En particular, la presente solicitud se relaciona con la codificación de video escalable (SVC), que incluye SVC para codificación de video avanzada (AVC) y SVC para codificación de video de alta eficiencia (HEVC), también conocida como HEVC escalable (SHVC), además de 3D y extensiones de vista múltiple para HEVC.

### 15 Antecedentes

20 Las capacidades de video digital pueden incorporarse en una amplia gama de dispositivos, que incluye televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, celular o teléfonos por radio satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de video en directo, y similares. Los dispositivos de video digital implementan procedimientos de codificación de video, tales como los descritos en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, codificación de video avanzada (AVC), el estándar (HEVC) actualmente en desarrollo, y 25 las extensiones de tales estándares. Los dispositivos de video pueden transmitir, recibir, codificar, decodificar y/o almacenar información de video digital más eficientemente mediante la implementación de tales tipos de codificación de video.

30 Los procedimientos de codificación de video incluyen la predicción espacial (intraimágenes) y/o la predicción temporal (interimágenes) para reducir o eliminar la redundancia inherente en las secuencias de video. Para la codificación de video basada en bloques, un segmento de video (por ejemplo, una trama de video o una porción de una trama de video) puede particionarse en bloques de video, que pueden, además, referirse a bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de video en un segmento intracodificado (I) de una imagen se codifican mediante el uso de la predicción espacial con respecto a las muestras de referencia en los 35 bloques vecinos de la misma imagen. Los bloques de video en un segmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden referirse a tramas y las imágenes de referencia pueden referirse como tramas de referencia.

40 La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan las 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 forma el bloque predictivo, y los datos residuales indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos 45 residuales. Para una compresión adicional, los datos residuales pueden transformarse del dominio de píxeles a un dominio de transformación, y dar como resultado coeficientes de transformación residuales, que luego pueden cuantificarse. Los coeficientes de transformación que se cuantifican, dispuestos inicialmente en una matriz bidimensional, pueden escanearse con el fin de producir un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación de entropía para lograr una compresión aún mayor.

50 Puede generarse un flujo de bits de codificación de vista múltiple al codificar vistas, por ejemplo, desde múltiples perspectivas. Se desarrollaron algunos estándares de video tridimensionales (3D) que hacen uso de aspectos de codificación de vista múltiple. Por ejemplo, diferentes vistas pueden transmitir vistas del ojo izquierdo y derecho para admitir video en 3D. Alternativamente, algunos procedimientos de codificación de video 3D pueden aplicar la denominada codificación de vista múltiple más profundidad. En la codificación de vista múltiple más profundidad, el 55 flujo de bits de video 3D puede contener no solo componentes de vista de textura, sino también componentes de vista de profundidad. Por ejemplo, cada vista puede comprender un componente de vista de textura y un componente de vista de profundidad.

60 Tech G y otros en MV-HEVC Draft Text 8 divulgan la señalización `profile_level_tier_idx[i]` para ( $i = 1; i < \text{NumOutputLayerSets}; i++$ ) en la sintaxis de extensión de VPS.

Sumario

65 En general, esta divulgación describe procedimientos y sistemas relacionados con la codificación de video, y más particularmente con HEVC. En los estándares actuales SHVC y MV-HEVC, el 0<sup>ésimo</sup> conjunto de capas de salida

(OLS) especificado por el conjunto de parámetros de video activo (VPS) se refiere al OLS que contiene solo la capa base de datos de video. Sin embargo, a menos que todo el flujo de bits contenga solo la capa base, la información de nivel y capa del 0<sup>ésimo</sup> OLS no se señaliza. Esto evita que el sistema de codificación de video funcione a un nivel óptimo en HEVC.

Los sistemas, procedimientos, y dispositivos de esta divulgación tienen cada uno varios aspectos innovadores, ninguno de los cuales es el único responsable de sus atributos convenientes divulgados en la presente memoria.

Un aspecto de la divulgación proporciona un procedimiento para codificar un flujo de bits de acuerdo con la reivindicación 1.

Otro aspecto de la divulgación proporciona un dispositivo para codificar de acuerdo con la reivindicación 2.

Otro aspecto de la divulgación proporciona un procedimiento para decodificar datos de video de acuerdo con la reivindicación 3.

Otro aspecto de la divulgación proporciona un dispositivo para decodificar datos de video de acuerdo con la reivindicación 4.

Otras características y ventajas de la presente divulgación deberían ser evidentes a partir de la siguiente descripción que ilustra, a modo de ejemplo, aspectos de la divulgación.

#### Descripción de los dibujos

Los detalles de las realizaciones de la presente divulgación, tanto en cuanto a su estructura como a su funcionamiento, pueden obtenerse en parte mediante el estudio de los dibujos adjuntos, en los que los números de referencia similares se refieren a partes similares, y en los que:

La Figura 1 es un diagrama de bloques funcional de un sistema de codificación de video;

La Figura 2 es un diagrama de bloques funcional del codificador de video de la Figura 1;

La Figura 3 es un diagrama de bloques funcional que ilustra el decodificador de video de la Figura 1;

La Figura 4 es un diagrama de flujo de señales que muestra un flujo de señales ejemplar entre un codificador de video y un decodificador de video;

La Figura 5 es un diagrama de flujo de un procedimiento para codificar un flujo de bits; y

La Figura 6 es un diagrama de flujo de un procedimiento para decodificar un flujo de bits.

#### Descripción detallada

La divulgación proporciona sistemas y procedimientos de codificación que pueden incluir una o más de varias mejoras en la codificación de video multicapa, incluida la señalización de información de interoperabilidad como perfil, capa y nivel para el 0<sup>ésimo</sup> conjunto de capas de salida (OLS). Como se usa en la presente memoria, la "0<sup>ésima</sup> capa" puede referirse generalmente a la capa base de un flujo de bits. En los estándares actuales SHVC y MV-HEVC, el 0<sup>ésimo</sup> OLS especificado por el VPS activo se refiere al OLS que contiene solo la capa base. Sin embargo, a menos que todo el flujo de bits contenga solo la capa base, la información de nivel y capa del 0<sup>ésimo</sup> OLS no se señaliza.

Varios aspectos de los sistemas, aparatos, y procedimientos novedosos se describen más completamente de aquí en adelante con referencia a los dibujos adjuntos. Sin embargo, esta divulgación puede llevarse a la práctica de muchas formas diferentes y no debe interpretarse como limitativa a ninguna estructura o función específica presentada durante esta divulgación. Más bien, estos aspectos se proporcionan para que esta divulgación se logre y complete, y transmita totalmente el ámbito de la divulgación para los expertos en la técnica. En base a las enseñanzas en la presente memoria un experto en la técnica debería apreciar que el ámbito de la divulgación pretende abarcar cualquier aspecto de los sistemas, aparatos y procedimientos novedosos divulgados en la presente memoria, ya sea implementado independientemente o combinado con cualquier otro aspecto de la divulgación. Por ejemplo, puede implementarse un aparato o puede practicarse un procedimiento mediante el uso de cualquier número de los aspectos expuestos en la presente memoria. Además, el ámbito de la divulgación pretende abarcar tal aparato o procedimiento el cual se practica mediante el uso de otra estructura, funcionalidad, o estructura y funcionalidad además o distinta de los diversos aspectos del conjunto de divulgación expuestos en la presente memoria. Debe entenderse que cualquier aspecto divulgado en la presente memoria puede llevarse a la práctica por uno o más elementos de una reivindicación.

#### Sistema de codificación de video

La Figura 1 es un diagrama de bloques funcional de un sistema de codificación de video. Un sistema de codificación de video ("sistema") 10 que puede utilizar procedimientos de acuerdo con los aspectos descritos en esta divulgación. Como se usa en la presente memoria, el término "codificador de video" se refiere de manera genérica tanto a los

codificadores de video como a los decodificadores de video. En esta divulgación, los términos "codificación de video" o "codificación" pueden referirse de manera genérica a la codificación de video o a la decodificación de video.

Como se muestra en la Figura 1, el sistema 10 incluye un dispositivo fuente 12 que puede generar los datos de video codificados a decodificarse en un momento posterior por un dispositivo de destino 14. El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, que incluyen ordenadores de escritorio, ordenadores portátiles (es decir, laptop), tabletas, módulos de conexión, teléfonos móviles tal como los denominados teléfonos "inteligentes", las denominadas almohadillas "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de videos o similares. En algunos casos, el dispositivo fuente 12 y el dispositivo de destino 14 pueden equiparse para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de video codificados para decodificarlos a través de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de video codificados desde el dispositivo fuente 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir al dispositivo fuente 12 transmitir los datos de video codificados de manera directa al dispositivo de destino 14 en tiempo real. El dispositivo fuente 12 puede modular los datos de video codificados de acuerdo con un estándar 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 de transmisión física. El medio de comunicación puede formar parte de una red en base a 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 enrutadores, conmutadores, estaciones base, o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo fuente 12 al dispositivo de destino 14.

Los datos codificados pueden enviarse desde la interfaz de salida 22 a un dispositivo de almacenamiento 34. De manera similar, puede accederse a los datos codificados desde el dispositivo de almacenamiento 34 mediante la interfaz de entrada. El dispositivo de almacenamiento 34 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de video codificados. En un ejemplo adicional, el dispositivo de almacenamiento 34 puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede contener el video codificado generado por el dispositivo fuente 12. El dispositivo de destino 14 puede acceder a los datos de video que se almacenan desde 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 los datos de video codificados y transmitir esos datos de video codificados al dispositivo de destino 14. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a la red (NAS), o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de video codificados a través de cualquier conexión de datos estándar, que incluye una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (por ejemplo, DSL, módem por cable, etc.), o una combinación de ambos que sea adecuada para acceder a los datos de video codificados almacenados en un servidor de archivos. La transmisión de datos de video codificados desde el dispositivo de almacenamiento 34 puede ser una transmisión continua, una transmisión de descarga, o una combinación de ambas.

Los procedimientos de esta divulgación no se limitan necesariamente a las aplicaciones o configuraciones inalámbricas. Los procedimientos pueden aplicarse a la codificación de video en soporte de cualquiera de una variedad de aplicaciones multimedia, tal como difusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video continuas, por ejemplo, a través de Internet, codificación de video digital para almacenamiento en un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para soportar la transmisión de video unidireccional o bidireccional para soportar aplicaciones tal como el video en directo, la reproducción de video, la difusión de video, y/o la telefonía de video.

En el ejemplo de la Figura 1, el dispositivo fuente 12 incluye una fuente de video 18, un codificador de video 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo fuente 12, la fuente de video 18 puede incluir una fuente tal como un dispositivo de captura de video, por ejemplo, una cámara de video, un archivo de video que contiene un video capturado previamente, una interfaz de alimentación de video para recibir el video de un proveedor de contenido de video, y/o un sistema de gráficos por ordenador para generar los datos de gráficos por ordenador como el video fuente, o una combinación de tales fuentes. Como un ejemplo, si la fuente de video 18 es una cámara de video, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, los procedimientos descritos en esta divulgación pueden ser aplicables a la codificación de video en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas.

El codificador de video 20 puede codificar el video capturado, precapturado, o generado por ordenador. Además, en algunas realizaciones, el decodificador de video 30 puede diferenciar entre un primer flujo de bits y un segundo flujo de bits en base al ID de conjunto de parámetros del primer flujo de bits y el ID de conjunto de parámetros único. En otros ejemplos, la interfaz de salida 22 del dispositivo fuente 12 puede transmitir los datos de video codificados directamente al dispositivo de destino 14 a través del enlace 16. Los datos de video codificados también (o de manera alternativa) pueden almacenarse en el dispositivo de almacenamiento 34 para un acceso posterior por el dispositivo de destino 14 u otros dispositivos, para la decodificación y/o reproducción.

El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de video 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 video codificados a través del enlace 16. Los datos de video codificados comunicados a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 34, pueden incluir una variedad de elementos de sintaxis generados por el codificador de video 20 para su uso por un decodificador de video, tal como el decodificador de video 30, en la decodificación de los datos de video. Tales elementos de sintaxis pueden incluirse con los datos de video codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

El dispositivo de visualización 32 puede integrarse con, o ser externo al, dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también configurarse para interactuar 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 muestra los datos de video decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico emisor de luz (OLED), u otro tipo de dispositivo de visualización.

El codificador de video 20 y el decodificador de video 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores adecuados, como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matriz de puertas lógicas programables en campo (FPGA), lógica discreta, software, hardware, firmware o cualquiera de sus combinaciones. Cuando los procedimientos se implementan de manera parcial en el software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador no transitorio adecuado y ejecutar las instrucciones en el hardware mediante el uso de uno o más procesadores para realizar los procedimientos de esta divulgación. Cada uno del codificador de video 20 y el decodificador de video 30 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo.

Los esfuerzos de la estandarización de HEVC se basan en un modelo evolutivo de un dispositivo de codificación de video denominado como el Modelo de Prueba HEVC (HM). El HM presupone diversas capacidades adicionales de los dispositivos de codificación de video con relación a los dispositivos existentes de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación de intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación de intrapredicción.

En general, el modelo de trabajo del HM describe que una trama o imagen de video puede dividirse en una secuencia de bloques de árbol o unidades de codificación más grandes (LCU) que incluyen tanto las muestras de luma como de croma. Un bloque de árbol tiene un propósito similar al de un macrobloque del estándar H.264. Un segmento incluye una serie de bloques de árbol consecutivos en el orden de codificación. Una trama o una imagen de video pueden dividirse en uno o más segmentos. Cada bloque de árbol puede dividirse en unidades de codificación (CU) de acuerdo con un árbol cuádruple. Por ejemplo, un bloque de árbol, como un nodo raíz del árbol cuádruple, puede dividirse en cuatro nodos hijo, y cada nodo hijo puede ser a su vez un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo no dividido final, como un nodo hoja del árbol cuádruple, comprende un nodo de codificación, es decir, un bloque de video codificado. Los datos de sintaxis asociados con un flujo de bits codificado pueden definir un número máximo de veces que un bloque de árbol puede dividirse, y definir un tamaño mínimo de los nodos de codificación.

Una CU incluye un nodo de codificación y las unidades de predicción (PU) y las unidades de transformación (TU) asociadas con el nodo de codificación. Un tamaño de la CU corresponde al tamaño del nodo de codificación y debe ser 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 superior. Cada CU puede contener una o más PU y una o más TU. Los datos de sintaxis asociados con una CU pueden describir, por ejemplo, la partición de la CU en una o más PU. Los modos de particionamiento pueden diferir entre si la CU se codifica en modo directo o en modo de salto, se codifica en modo de intrapredicción, o se codifica en modo de interpredicción. Las PU pueden dividirse para que sean de forma no cuadrada. Los datos de sintaxis asociados con una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuádruple. Una TU puede ser de forma cuadrada o no cuadrada.

El estándar HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. Las TU pueden dimensionarse en base al tamaño de las PU dentro de una CU determinada definida para una LCU

particionada, aunque esto puede no ser siempre así. Las TU pueden ser del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas mediante el uso de una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos hoja del RQT pueden referirse como unidades de transformación (TU). Los valores de diferencia de píxeles asociados con las TU pueden transformarse para producir coeficientes de transformación, que pueden cuantificarse.

En general, una PU incluye los datos relacionados con el procedimiento de predicción. Por ejemplo, cuando la PU se codifica dentro de modos, la PU puede incluir los datos que describen un modo de intrapredicción para la PU. Como otro ejemplo, cuando la PU se codifica entre modos, la PU puede incluir los 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, Lista 0, Lista 1, o Lista C) para el vector de movimiento.

En general, una TU se utiliza para los procedimientos de transformación y cuantificación. Una CU dada que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Después de la predicción, el codificador de video 20 puede calcular los valores residuales correspondientes a la PU. Los valores residuales comprenden valores de diferencia de píxeles que pueden transformarse en los coeficientes de transformación, cuantificarse y explorarse mediante el uso de las TU para producir los coeficientes de transformación serializados para la codificación por entropía. Como se usa en la presente memoria, el término "bloque de video" puede referirse generalmente a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede utilizar el término "bloque de video" para referirse a un bloque de árbol, es decir, LCU, o CU, que incluye un nodo de codificación y las PU y las TU.

El codificador de video 20 puede generar un flujo de bits que incluye datos de video codificados. El flujo de bits puede incluir una serie de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL puede incluir un encabezado de la unidad NAL y puede encapsular una carga útil de secuencia de bytes sin procesar (RBSP). El encabezado de la unidad NAL puede incluir un elemento de sintaxis que indica un código de tipo de unidad NAL. El código de tipo de unidad NAL que se especifica por el encabezado de la unidad NAL de una unidad NAL indica el tipo de unidad NAL. Las unidades NAL del flujo de bits pueden incluir unidades NAL de capa de codificación de video (VCL) y unidades NAL sin VCL. Las unidades NAL de VCL pueden incluir segmentos codificados de imágenes.

Una unidad NAL sin VCL puede incluir un VPS, un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), SEI u otros tipos de datos. Un VPS es una estructura de sintaxis que puede contener elementos de sintaxis que se aplican a cero o más secuencias de video codificadas completas. Un SPS es una estructura de sintaxis que puede contener elementos de sintaxis que se aplican a cero o más secuencias de video codificadas completas. Un solo VPS puede ser aplicable a varios SPS. Un PPS es una estructura de sintaxis que puede contener elementos de sintaxis que se aplican a cero o más imágenes codificadas completas. Un único SPS puede ser aplicable a varios PPS. Pueden formarse varios aspectos del VPS, SPS y PPS, en general, según lo definido por el estándar HEVC. El codificador de video 20 puede usar mensajes SEI para incluir, en el flujo de bits, metadatos que no son necesarios para la decodificación correcta de los valores de muestra de imágenes. Sin embargo, el decodificador de video 30 u otros dispositivos pueden usar los metadatos incluidos en los mensajes SEI para varios otros propósitos. Por ejemplo, el decodificador de video 30 puede utilizar los metadatos en los mensajes SEI para la sincronización de la salida de imágenes, la visualización de imágenes, la detección de pérdidas y la ocultación de errores.

## Sintaxis de alto nivel

Las unidades NAL pueden contener elementos de sintaxis codificados por el codificador de video 20 y transmitidos al decodificador de video 30. Los elementos de sintaxis que describen la estructura del flujo de bits o proporcionan información que se aplica a múltiples imágenes o a múltiples regiones de bloques codificados dentro de una imagen, como los conjuntos de parámetros (por ejemplo, SPS, VPS, PPS, etc.), sintaxis de gestión de imágenes de referencia, y los mensajes SEI, se conocen como parte de "sintaxis de alto nivel" (HLS) de HEVC.

En particular, HEVC emplea los siguientes aspectos, relevantes para esta divulgación: *Estructura del conjunto de parámetros*: Los conjuntos de parámetros contienen información que puede compartirse para la decodificación de varias regiones del video decodificado. La estructura del conjunto de parámetros proporciona un mecanismo robusto para transmitir datos que son esenciales para el procedimiento de decodificación. *Información de mejora complementaria* (SEI): La sintaxis incluye soporte para varios tipos de metadatos conocidos como SEI. Dichos datos proporcionan información sobre la sincronización de las imágenes de video, la interpretación adecuada del espacio de color utilizado en la señal de video, información de empaquetado de tramas estereoscópicas en 3D, otra información de sugerencia de visualización, etc.

La Figura 2 es un diagrama de bloques funcional del codificador de video de la Figura 1. El codificador de video 20 puede incluir un módulo de partición 35 configurado para recibir datos de video capturados y dividir los datos de video en bloques de video. El módulo de partición 35 puede dividir los datos de video en segmentos, mosaicos u otras unidades más grandes. El módulo de división 35 también puede dividir bloques de video, por ejemplo, de acuerdo con una estructura de cuatro árboles de LCU y CU. El codificador de video 20 puede dividir los segmentos en múltiples bloques de video (y posiblemente en conjuntos de bloques de video denominados mosaicos).

El codificador de video 20 también puede tener un módulo de predicción 41 acoplado operativamente al módulo de partición 35 y configurado para recibir los datos de video particionados. El módulo de predicción 41 puede seleccionar uno de una pluralidad de modos de codificación posibles, tal como uno de una pluralidad de modos de intracodificación o uno de una pluralidad de modos de intercodificación, para el bloque de video actual en base a resultados de error (por ejemplo, tasa de codificación y el nivel de distorsión). El módulo de predicción 41 puede permitir que el codificador de video 20 realice la intracodificación y la intercodificación de los bloques de video dentro de los segmentos de video. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el video dentro de una trama o imagen de video determinado. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el video dentro de tramas o imágenes adyacentes de una secuencia de video. El intramodo (modo I) puede referirse a cualquiera de diversos modos de compresión basados en el espacio. Los intermodos, tal como la predicción unidireccional (modo P) o la bipredicción (modo B), pueden referirse a cualquiera de diversos modos de compresión basados en el tiempo.

El módulo de predicción 41 puede incluir un módulo de estimación de movimiento 42, un módulo de compensación de movimiento 44 y un módulo de intrapredicción 46. Cada uno de estos componentes puede conectarse o integrarse operativamente en uno o más procesadores dentro del codificador de video 20. Aunque se analizan por separado con fines conceptuales, el módulo de estimación de movimiento 42, el módulo de compensación de movimiento 44 y el módulo de intrapredicción 46 pueden encontrarse altamente integrados.

El módulo de estimación del movimiento 42 puede configurarse para determinar el modo de interpredicción para un segmento de video de acuerdo con un patrón predeterminado para una secuencia de video. Tal interpredicción o codificación interpredictiva del bloque de video actual en relación con uno o más bloques predictivos en una o más imágenes de referencia puede proporcionar compresión temporal. El patrón predeterminado puede designar segmentos de video en la secuencia como segmentos P, segmentos B o segmentos GPB. La estimación de movimiento, como se usa en la presente memoria, puede referirse generalmente al procedimiento de generar vectores de movimiento, que estiman el movimiento para bloques de video. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de video dentro de una imagen o trama de video actual con relación a un bloque predictivo dentro de una trama de referencia o imagen de referencia. Las tramas de referencia pueden almacenarse en una memoria de imágenes de referencia 64 acoplada operativamente al módulo de predicción 41. La memoria de la trama de referencia 64. La memoria de la trama de referencia 64 puede almacenar bloques de codificación reconstruidos después de que un módulo de filtro 63 realice operaciones de desbloqueo en los bloques de codificación reconstruidos. Por tanto, la memoria de la trama de referencia 64 también puede funcionar como una memoria intermedia de imágenes decodificadas (DPB).

Un bloque predictivo es un bloque que se encuentra que coincide estrechamente con la PU del bloque de video a codificar en términos de diferencia de píxeles, que puede determinarse mediante la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD), o mediante otras métricas de diferencia. En algunos ejemplos, el módulo de predicción de movimiento ("módulo de predicción") 41 puede calcular valores para posiciones de píxeles subenteros de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el módulo de predicción 41 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxeles fraccionarias de la imagen de referencia. Por lo tanto, el módulo de estimación de movimiento 42 puede realizar una búsqueda de movimiento con relación a las posiciones de píxeles completas y a las posiciones de píxeles fraccionarias y generar un vector de movimiento con precisión de píxel fraccionaria.

El módulo de estimación de movimiento 42 puede calcular un vector de movimiento para una PU de un bloque de video en un segmento intercodificado al comparar la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse de una primera lista de imágenes de referencia (Lista 0) o de una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. El módulo de estimación de movimiento 42 puede enviar el vector de movimiento calculado al módulo de compensación de movimiento 44 y un módulo de codificación de entropía 56.

La compensación de movimiento puede implicar buscar o generar el bloque predictivo en base al vector de movimiento determinado por la estimación de movimiento. Esto puede incluir interpolaciones con precisión de subpíxeles. Al recibir el vector de movimiento para la PU del bloque de video actual, el módulo de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El módulo de compensación de movimiento 44 también puede generar los elementos de sintaxis asociados con los bloques de video y el segmento de video para su uso por el decodificador de video 30 al decodificar los bloques de video del segmento de video.

El módulo de intrapredicción 46 del módulo de predicción 41 puede realizar la codificación intrapredictiva del bloque de video actual en relación a uno o más bloques vecinos en la misma trama o segmento para proporcionar la compresión espacial. El módulo de intrapredicción 46 puede intraprededir un bloque actual, como una alternativa a la interpredicción que se realiza mediante el módulo de estimación del movimiento 42 y el módulo de compensación del movimiento 44, como se describió anteriormente. En particular, el módulo de intrapredicción 46 puede determinar un modo de intrapredicción para usar para codificar un bloque actual. En algunos ejemplos, el módulo de intrapredicción 46 puede codificar un bloque actual mediante el uso de varios modos de intrapredicción, por ejemplo, durante pasadas de codificación separadas. Por ejemplo, el módulo de intrapredicción 46 puede calcular valores de tasa de distorsión al usar un análisis de tasa de distorsión para varios modos de intrapredicción probados, y seleccionar el modo de intrapredicción que tiene las mejores características de tasa de distorsión entre los modos probados. El análisis de la tasa de distorsión puede determinar una cantidad de distorsión (o error) entre un bloque codificado y uno original, el bloque no codificado que se codificó para producir el bloque codificado, así como también una tasa de bits (es decir, una cantidad de bits) que se usa para producir el bloque codificado. El módulo de intrapredicción 46 puede calcular las relaciones de las distorsiones y tasas para varios bloques codificados para determinar qué modo de intrapredicción exhibe el mejor valor de tasa de distorsión para el bloque.

El módulo de intrapredicción 46 puede proporcionar información indicativa del modo de intrapredicción seleccionado para el bloque al módulo de codificación de entropía 56. El módulo de codificación de entropía 56 puede codificar la información que indica el modo de intrapredicción seleccionado. El codificador de video 20 puede incluir datos de configuración en el flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índice en modo de intrapredicción y una pluralidad de tablas de índice en modo de intrapredicción modificadas (también denominadas, tablas de asignación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índice de modo de intrapredicción y una tabla de índice de modo de intrapredicción modificada para usar en cada uno de los contextos.

Después de que el módulo de predicción 41 genera el bloque predictivo para el bloque de video actual, ya sea a través de intrapredicción o interpredicción, el codificador de video 20 forma un bloque de video residual al restar el bloque predictivo del bloque de video actual. Este procedimiento puede resultar en valores de diferencia de píxeles. Los valores de diferencia de píxeles forman los datos residuales para el bloque, y pueden incluir componentes de diferencia de luma y croma. Un sumador 50, acoplado operativamente al módulo de predicción 41 y al módulo de partición 35, representa el componente o componentes que pueden configurarse para realizar esta operación de resta.

Los datos de video residuales en el bloque residual pueden incluirse en una o más TU y aplicarse a un módulo de transformación 52 acoplado de forma operativa al sumador 50. El módulo de transformación 52 puede transformar los datos de video residuales en coeficientes de transformación residual mediante el uso de una transformación, tal como una transformación de coseno discreta (DCT) o una transformación conceptualmente similar. El módulo de transformación 52 puede convertir los datos de video residuales de un dominio de píxeles a un dominio de transformación, tal como un dominio de frecuencia. El módulo de predicción 41 puede proporcionar el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloques residuales. El bloque intracodificado o intercodificado resultante puede proporcionarse además como un sumador 62 para reconstruir el bloque codificado para su uso como imagen de referencia.

El codificador de video 20 también puede incluir el módulo de filtro 63 acoplado operativamente al sumador 62. El módulo de filtro 63 puede representar uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de bucle adaptativo (ALF) y un filtro de compensación adaptativa de muestra (SAO). Aunque el módulo de filtro 63 se muestra en la Figura 2 como un filtro en bucle, en otras configuraciones, el módulo de filtro 63 puede implementarse como un filtro de bucle posterior. El módulo de filtro 63 puede proporcionar imágenes de referencia a la memoria de imágenes de referencia 64.

El módulo de transformación 52 puede enviar los coeficientes de transformación resultantes al módulo de cuantificación 54. El módulo de cuantificación 54 puede cuantificar los coeficientes de transformación para reducir aún más la tasa de bits. El procedimiento de cuantificación puede reducir la profundidad de bits que se asocia con algunos o todos los coeficientes. El grado de cuantificación puede modificarse mediante el ajuste de un parámetro de cuantificación. En algunos ejemplos, el módulo de cuantificación 54 puede luego realizar una exploración de la matriz que incluye los coeficientes de transformación cuantificados. En algunas realizaciones, el módulo de codificación de entropía 56 puede realizar la exploración.

Después de la cuantificación, el módulo de codificación por entropía 56 puede codificar por entropía los coeficientes de transformación cuantificados. Por ejemplo, el módulo de codificación por entropía 56 puede realizar codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto en base a sintaxis (SBAC), codificación por entropía de partición de intervalo de probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de video 20 también puede codificar por entropía los elementos de sintaxis asociados con los datos de video codificados para su uso por el decodificador de video 30 al decodificar los datos de video.



Después de la codificación de entropía por el módulo de codificación de entropía 56, el flujo de bits codificado puede transmitirse al decodificador de video 30. El flujo de bits también puede archivarse para su posterior transmisión o recuperación por el decodificador de video 30. El módulo de codificación por entropía 56 también puede codificar por entropía los vectores de movimiento y los otros elementos de sintaxis para el segmento de video actual que se codifica.

El codificador de video 20 también puede incluir un módulo de cuantificación inversa 58 acoplado operativamente al módulo de codificación de entropía 56. Un módulo de transformación inversa 60 también puede acoplarse operativamente al módulo de cuantificación inversa 58 y al sumador 62. El módulo de cuantificación inversa 58 y el módulo de transformación inversa 60 pueden aplicar la cuantificación inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles para su uso posterior como bloque de referencia de una imagen de referencia. El módulo de compensación de movimiento 44 puede calcular un bloque de referencia al añadir el bloque residual a un bloque predictivo de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El sumador 62 puede añadir el bloque residual reconstruido al bloque de predicción compensado por el movimiento producido por el módulo de compensación de movimiento 44 para producir un bloque de referencia para almacenar en la memoria de imágenes de referencia 64. El bloque de referencia puede usarse por la unidad de estimación de movimiento 42 y por el módulo de compensación de movimiento 44 como un bloque de referencia para interpredecir un bloque en una trama o imagen de video subsecuente.

En algunos ejemplos, el codificador de video 20 genera un flujo de bits. El flujo de bits generado por el codificador de video 20 puede incluir múltiples puntos de operación que pueden extraerse como subflujos de bits del flujo de bits. Los puntos de operación pueden incluir, por ejemplo, múltiples capas y/o vistas, así como múltiples tasas de cuadro. El codificador de video 20 puede codificar información indicativa de los puntos de operación de salida en el VPS. Como se usa en la presente memoria, un punto de operación de salida se refiere a un punto de operación para el cual se especifican las capas de salida (es decir, el conjunto de capas que se generarán). En algunos ejemplos, para cada uno de los puntos de operación que el codificador de video 20 señala en el VPS asociado con el flujo de bits, una estructura de sintaxis de punto de operación especifica un conjunto de identificadores de capa (ID) utilizados para identificar las unidades NAL en el flujo de bits que pertenecen a un subflujo de bits de un punto de operación dado. De esta manera, las unidades NAL que forman el subflujo de bits del punto de operación dado pueden extraerse del flujo de bits original en base a los identificadores de capa de las unidades NAL.

La Figura 3 es un diagrama de bloques que ilustra el decodificador de video de la Figura 1. Con fines de explicación, esta divulgación describe el decodificador de video 30 en el contexto de codificación HEVC. Sin embargo, esta divulgación puede ser aplicable a otros estándares o procedimientos de codificación.

El decodificador de video 30 puede incluir un módulo de decodificación por entropía 70, un módulo de predicción 71, un módulo de cuantificación inversa 76, un módulo de transformación inversa 78, un sumador 80, un módulo de filtro 84 y una memoria intermedia de imagen decodificada (DPB) 82. El módulo de predicción 71 puede incluir un módulo de compensación de movimiento 72 y un módulo de intrapredicción 74. En otros ejemplos, el decodificador de video 30 puede incluir más, menos, o diferentes componentes funcionales.

Una memoria intermedia de imágenes codificadas (CPB) 90 puede recibir y almacenar los datos de video codificados (por ejemplo, las unidades NAL) de un flujo de bits. El módulo de decodificación por entropía 70 puede recibir unidades NAL de CPB 90 y analizar las unidades NAL para decodificar elementos de sintaxis. El módulo de decodificación por entropía 70 puede decodificar por entropía los elementos de sintaxis codificados por entropía en las unidades NAL. El módulo de predicción 71, el módulo de cuantificación inversa 76, el módulo de transformación inversa 78, el sumador 80 y el módulo de filtro 84 pueden generar datos de video decodificados en base a los elementos de sintaxis extraídos del flujo de bits.

Las unidades NAL del flujo de bits pueden incluir las unidades NAL del segmento codificado. Como parte de la decodificación del flujo de bits, el módulo de decodificación por entropía 70 puede extraer y decodificar por entropía los elementos de sintaxis de las unidades NAL del segmento codificado. Cada uno de los segmentos codificados puede incluir un encabezado del segmento y los datos del segmento. El encabezado del segmento puede contener los elementos de sintaxis pertenecientes a un segmento. Los elementos de sintaxis en el encabezado del segmento pueden incluir un elemento de sintaxis que identifica un PPS asociado con una imagen que contiene el segmento.

Además de decodificar los elementos de sintaxis del flujo de bits, el decodificador de video 30 puede realizar una operación de reconstrucción en una CU no particionada. Para realizar la operación de reconstrucción en una CU no particionada, el decodificador de video 30 puede realizar una operación de reconstrucción en cada TU de la CU. Al realizar la operación de reconstrucción para cada TU de la CU, el decodificador de video 30 puede reconstruir los bloques residuales de la CU.

Como parte de la realización de una operación de reconstrucción en una TU de una CU, el módulo de cuantificación inversa 76 puede cuantificar de forma inversa, es decir, descuantificar, los bloques de coeficientes asociados con la TU. El módulo de cuantificación inversa 76 puede usar un valor de parámetro de cuantificación (QP) asociado con la

CU de la TU para determinar un grado de cuantificación y, de igual forma, un grado de cuantificación inversa para que se aplique el módulo de cuantificación inversa 76. Es decir, la relación de compresión, es decir, la relación entre el número de bits utilizados para representar la secuencia original y la comprimida, puede controlarse al ajustar el valor del QP utilizado al cuantificar los coeficientes de transformación. La relación de compresión también puede depender del procedimiento de codificación por entropía empleado.

Después de que el módulo de cuantificación inversa 76 cuantifique inversamente un bloque de coeficientes, el módulo de transformación inversa 78 puede aplicar una o más transformaciones inversas al bloque de coeficientes para generar un bloque residual asociado con la TU. Por ejemplo, el módulo de transformación inversa 78 puede aplicar una DCT inversa, una transformación entera inversa, una transformación direccional inversa, u otra transformación inversa al bloque de coeficientes.

Si se codifica una PU mediante el uso de la intrapredicción, el módulo de intrapredicción 74 puede realizar la intrapredicción para generar los bloques predictivos para la PU. El módulo de intrapredicción 74 puede utilizar un modo de intrapredicción para generar los bloques predictivos de luma, Cb y Cr para la PU en base a los bloques de predicción de las PU vecinas espacialmente. El módulo de intrapredicción 74 puede determinar el modo de intrapredicción para la PU en base a uno o más elementos de sintaxis decodificados del flujo de bits.

El módulo de predicción 71 puede construir una primera lista de imágenes de referencia (RefPicList0) y una segunda lista de imágenes de referencia (RefPicList1) en base a los elementos de sintaxis extraídos del flujo de bits. Además, si se codifica una PU mediante el uso de la interpredicción, el módulo de decodificación por entropía 70 puede extraer la información de movimiento para la PU. El módulo de compensación de movimiento 72 puede determinar, en base a la información de movimiento de la PU, una o más regiones de referencia para la PU. El módulo de compensación de movimiento 72 puede generar bloques predictivos de luma, Cb y Cr para la PU basados en bloques de muestras en el uno o más bloques de referencia para la PU.

El sumador 80 puede utilizar los bloques de transformación de luma, Cb y Cr asociados con las TU de una CU y los bloques predictivos de luma, Cb y Cr de las PU de la CU, es decir, los datos de intrapredicción o los datos de interpredicción, según corresponda, para reconstruir los bloques de codificación de luma, Cb y Cr de la CU. Por ejemplo, el sumador 80 puede añadir muestras de los bloques de transformación de luma, Cb y Cr a las muestras correspondientes de los bloques predictivos de luma, Cb y Cr para reconstruir los bloques de codificación de luma, Cb y Cr de la CU.

El módulo de filtro 84 puede ser un filtro de desbloqueo, por ejemplo, y realizar una operación de desbloqueo para reducir los artefactos de bloqueo asociados con los bloques de codificación de luma, Cb y Cr de la CU. El decodificador de video 30 puede entonces almacenar los bloques de codificación de luma, Cb y Cr de la CU en la memoria intermedia de imágenes decodificadas 82. La memoria intermedia de imágenes decodificadas 82 puede proporcionar imágenes de referencia para la compensación de movimiento subsecuente, la intrapredicción y la presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la Figura 1. Por ejemplo, el decodificador de video 30 puede realizar operaciones de intrapredicción o interpredicción en PU de otras CU en base a los bloques de luma, Cb y Cr en la memoria intermedia de imágenes decodificadas 82. Por lo tanto, el decodificador de video 30 puede decodificar, a partir del flujo de bits, los niveles de coeficiente de transformación del bloque de coeficiente de luma significativo, cuantificar inversamente los niveles de coeficiente de transformación, aplicar una transformación a los niveles de coeficiente de transformación para generar un bloque de transformación, generar, basado, al menos en parte, del bloque de transformación, un bloque de codificación, y dar salida al bloque de codificación para su visualización.

#### Orden de conjuntos de parámetros

Un RBSP es una estructura de sintaxis que contiene un número entero de bytes que se encapsula en una unidad NAL y que está vacío o tiene la forma de una cadena de bits de datos que contienen elementos de sintaxis.

Un PPS RBSP incluye parámetros a los que puede hacerse referencia mediante las unidades NAL del segmento de corte codificado de una o más imágenes codificadas. Cada PPS RBSP se considera inicialmente no activo para la capa base al comienzo de la operación del procedimiento de decodificación. A lo sumo, un PPS RBSP se considera activo para la capa base en un momento dado durante la operación del procedimiento de decodificación, y la activación de cualquier PPS RBSP particular para la capa base da como resultado la desactivación del PPS RBSP previamente activo para la capa base (si la hubiera).

Cuando un PPS RBSP no está activo para la capa base y se refiere por una unidad NAL de segmento de corte codificado con nuh\_layer\_id igual a 0, se activa para la capa base. Este PPS RBSP se denomina PPS RBSP activo para la capa base hasta que se desactiva mediante la activación de otro PPS RBSP para la capa base. Un PPS RBSP, con ese valor particular de pps\_pic\_parameter\_set\_id, estará disponible para el procedimiento de decodificación antes de su activación, incluido en, al menos, una unidad de acceso con TemporalId menor o igual al TemporalId de la unidad PPS NAL o proporcionado a través de medios externos, y la unidad PPS NAL que contiene el PPS RBSP tendrá nuh\_layer\_id igual a 0.

Cualquier unidad PPS NAL que contenga el valor de `pps_pic_parameter_set_id` para la PPS RBSP activa para una imagen codificada (y, en consecuencia, para la capa que contiene la imagen codificada) tendrá el mismo contenido que la PPS RBSP activa para la imagen codificada, a menos que siga la última unidad VCL NAL de la imagen codificada y precede a la primera unidad VCL NAL de otra imagen codificada.

Una SPS RBSP incluye parámetros a los que puede hacerse referencia mediante una o más PPS RBSP o una o más unidades SEI NAL que contienen un mensaje SEI de conjuntos de parámetros activos. Cada SPS RBSP se considera inicialmente no activo para la capa base al comienzo de la operación del procedimiento de decodificación. Como máximo, una SPS RBSP se considera activa para la capa base en cualquier momento durante la operación del procedimiento de decodificación, y la activación de cualquier SPS RBSP en particular para la capa base da como resultado la desactivación de la SPS RBSP previamente activa para la capa base (si la hubiera).

Cuando un SPS RBSP aún no está activo para la capa base y se hace referencia a él mediante la activación de un PPS RBSP para la capa base o, cuando `vps_base_layer_internal_flag` es igual a 1 y `vps_base_layer_available_flag` es igual a 1, se lo denomina una unidad SEI NAL que contiene un mensaje SEI de conjuntos de parámetros activos, se activa para la capa base. Esta SPS RBSP se denomina SPS RBSP activa para la capa base hasta que se desactiva mediante la activación de otra SPS RBSP para la capa base. Un SPS RBSP, con ese valor particular de `sps_seq_parameter_set_id`, estará disponible para el procedimiento de decodificación antes de su activación, incluido en, al menos, una unidad de acceso con `TemporalId` igual a 0 o proporcionado a través de medios externos, y la unidad SPS NAL que contiene el SPS RBSP tendrá `nuh_layer_id` igual a 0. Un SPS RBSP activado para la capa base permanecerá activo durante toda la secuencia de video codificada (CVS).

Cualquier unidad SPS NAL con `nuh_layer_id` igual a 0 que contenga el valor de `sps_seq_parameter_set_id` para el SPS RBSP activo para la capa base para un CVS deberá tener el mismo contenido que el SPS RBSP activo para la capa base para el CVS, a menos que siga a la última unidad de acceso del CVS y precede a la primera unidad VCL NAL y la primera unidad SEI NAL que contiene un mensaje SEI de conjuntos de parámetros activos (cuando está presente) de otro CVS.

Un VPS RBSP incluye parámetros a los que puede hacer referencia una o más SPS RBSP o una o más unidades SEI NAL que contienen un mensaje SEI de conjuntos de parámetros activos. Cada VPS RBSP se considera inicialmente no activo al inicio de la operación del procedimiento de decodificación. Como máximo, un VPS RBSP se considera activo en cualquier momento dado durante la operación del procedimiento de decodificación, y la activación de cualquier VPS RBSP particular da como resultado la desactivación del VPS RBSP previamente activo (si lo hubiera).

Cuando un VPS RBSP (con un valor particular de `vps_video_parameter_set_id`) aún no está activo y se hace referencia a él mediante la activación de un SPS RBSP para la capa base, o se lo conoce por una unidad SEI NAL que contiene un mensaje SEI de conjuntos de parámetros activos, está activado. Este VPS RBSP se denomina VPS RBSP activo hasta que se desactiva mediante la activación de otro VPS RBSP. Un VPS RBSP, con ese valor particular de `vps_video_parameter_set_id`, estará disponible para el procedimiento de decodificación antes de su activación, incluido en, al menos, una unidad de acceso con `TemporalId` igual a 0 o proporcionado a través de medios externos, y la unidad VPS NAL que contiene el VPS RBSP tendrá `nuh_layer_id` igual a 0. Un VPS RBSP activado permanecerá activo durante todo el CVS.

Cualquier unidad VPS NAL que contenga el valor de `vps_video_parameter_set_id` para el VPS RBSP activo para un CVS deberá tener el mismo contenido que el del VPS RBSP activo para el CVS, a menos que siga a la última unidad de acceso del CVS y preceda a la primera unidad VCL NAL, la primera unidad SPS NAL y la primera unidad SEI NAL que contienen un mensaje SEI de conjuntos de parámetros activos (cuando está presente) de otro CVS.

Todas las restricciones que se expresan en la relación entre los valores de los elementos de sintaxis y los valores de las variables derivadas de esos elementos de sintaxis en los VPS, SPS y PPS y otros elementos de sintaxis son expresiones de restricciones que se aplican solo al VPS RBSP activo, el SPS RBSP activo para la capa base y el PPS RBSP activo para la capa base. Si hay algún VPS RBSP, SPS RBSP y PPS RBSP que nunca se activa en el flujo de bits, sus elementos de sintaxis deberán tener valores que se ajusten a las restricciones especificadas si se activara por referencia en un flujo de bits conformes.

Durante la operación del procedimiento de decodificación, los valores de los parámetros del VPS RBSP activo, el SPS RBSP activo para la capa base y el PPS RBSP activo para la capa base se consideran vigentes. Para la interpretación de los mensajes SEI, los valores del VPS RBSP activo, el SPS RBSP activo para la capa base y el PPS RBSP activo para la capa base para la operación del procedimiento de decodificación para las unidades VCL NAL de la imagen codificada con `nuh_layer_id` igual a 0 en la misma unidad de acceso se consideran en vigor a menos que se especifique lo contrario en la semántica del mensaje SEI.

Puntos de operación, puntos de operación de salida, conjuntos de capas y conjuntos de capas de salida

Como se usa en la presente memoria, un punto de operación es un flujo de bits creado a partir de otro flujo de bits mediante la operación del procedimiento de extracción de subflujo de bits con otro flujo de bits, un ID temporal más alto de destino y una lista de identificadores de la capa de destino como entradas. Más específicamente, los "puntos de operación" son subflujos de bits que pueden extraerse de un flujo de bits original que es escalable temporalmente y/o con múltiples capas y/o vistas. Es decir, un punto de operación es un flujo de bits que se crea a partir de otro flujo de bits mediante la operación del procedimiento de extracción del subflujo de bits con el otro flujo de bits, un ID temporal más alto de destino y una lista de identificadores de la capa de destino como entradas. El flujo de bits incluye una o más vistas, incluidas las vistas del punto de operación. El ID temporal más alto de destino define el subconjunto temporal. La lista de capas de destino incluye un conjunto de capas: el conjunto de capas que se decodificarán para un punto de operación y el conjunto de capas que se generarán para un punto de operación de salida. Un punto de operación de salida se asocia, entre otras cosas, con un conjunto de capas de salida, es decir, las capas identificadas en el conjunto de capas de salida, que son las capas que se van a producir. En general, cada flujo de bits puede contener uno o más conjuntos de capas de salida. Cada conjunto de capas de salida puede contener uno o más puntos de operación de salida. Se realiza una prueba de conformidad del flujo de bits para cada punto de operación de salida para garantizar que el subflujo de bits correspondiente es un flujo de bits conforme.

El término "conjunto de capas" se utiliza para hacer referencia a un conjunto de capas a decodificar. Un "conjunto de capas" es similar a un punto de operación excepto que un conjunto de capas no se asocia con un valor particular de ID temporal. Más bien, las subcapas temporales presentes en las capas pertenecen todas al conjunto de capas. De manera similar, puede decirse que un conjunto de capas se asocia con el ID temporal más alto posible. En consecuencia, las subcapas temporales presentes en las capas pertenecen todas al conjunto de capas. Un conjunto de capas puede derivarse de un flujo de bits de la misma manera que un punto de operación puede derivarse de un flujo de bits, el ID temporal es más alto el valor más alto posible.

El término "conjunto de capas de salida" se refiere a un conjunto de capas para las que se especifican las capas de salida. Por lo tanto, mientras que el conjunto de capas a decodificar se especifica como un conjunto de capas, el conjunto de capas a generar se especifica para un conjunto de capas de salida. En algunos ejemplos, el conjunto de capas de salida identifica el conjunto de capas correspondiente al incluir un índice al conjunto de capas correspondiente, donde el conjunto de capas correspondiente identifica las capas que se van a decodificar. En otros ejemplos, el conjunto de capas de salida identifica el conjunto de capas correspondiente de alguna otra manera.

Las capas del conjunto de capas que no se destinan a la salida aún pueden decodificarse, por ejemplo, cuando se utilizan para la predicción entre capas (o entre vistas). Las capas que se destinan a la salida son un subconjunto de las capas que se van a decodificar: una, algunas o todas las capas que se van a decodificar pueden destinarse a la salida. Debe entenderse que el término "subconjunto" no significa necesariamente que la capa que se va a generar es un subconjunto estricto de las capas que se van a decodificar. Es decir, en algunos ejemplos, las capas que se van a decodificar y la capa que se va a generar son las mismas. Por tanto, la distinción entre un conjunto de capas de salida y un punto de operación de salida es similar a la distinción entre un conjunto de capas y un punto de operación como se describe anteriormente.

Los subflujos de bits pueden extraerse del flujo de bits en base a los valores de los identificadores de capa (que identifican las capas para el punto de operación de salida) y los identificadores de subcapa temporal (que identifican el subconjunto temporal para el punto de operación de salida) que identifican el punto de operación de salida del flujo de bits. Los puntos de operación de salida pueden señalarse en un VPS dentro del flujo de bits. Para cada uno de los puntos de operación, una estructura de sintaxis de punto de operación especifica un conjunto de identificadores de capa utilizados para identificar unidades de capa de abstracción de red (NAL) en el flujo de bits que pertenecen a un subflujo de bits de un punto de operación dado. De esta manera, las unidades NAL que forman el subflujo de bits del punto de operación de salida dado pueden extraerse del flujo de bits original en base a los identificadores de capa de las unidades NAL y el ID temporal más alto asociado con el punto de operación de salida. Las unidades NAL son los componentes de los datos de video codificados; los datos de video codificados se organizan en unidades NAL.

Sintaxis de perfil, capa, nivel

La estructura de sintaxis `profile_tier_level()` es una estructura de sintaxis que proporciona información de sintaxis relacionada con perfiles, capas y niveles. Los perfiles, capas y niveles especifican diferentes puntos de conformidad para implementar el estándar HEVC o un estándar de extensión HEVC (por ejemplo, SHVC, MV-HEVC) de manera que sea interoperable entre varias aplicaciones que pueden tener requisitos funcionales similares. En algunos ejemplos, un perfil puede definir un conjunto de herramientas de codificación o algoritmos que un códec puede utilizar para generar un flujo de bits conforme. Un perfil es un subconjunto de toda la sintaxis del flujo de bits especificado por un estándar particular, HEVC, por ejemplo. Dentro de los límites impuestos por la sintaxis de un perfil dado, todavía es posible requerir una variación muy grande en el rendimiento de los codificadores y decodificadores en dependencia de los valores tomados por los elementos de sintaxis en el flujo de bits, como el tamaño especificado de las imágenes decodificadas. En muchas aplicaciones, puede que no sea práctico ni económico implementar un decodificador (por ejemplo, el decodificador de video 30) capaz de manejar todos los usos hipotéticos de la sintaxis dentro de un perfil particular. Por lo tanto, las "capas" y los "niveles" también se

especifican dentro de cada perfil. Un nivel de una capa es un conjunto específico de restricciones impuestas a los valores de los elementos de sintaxis en el flujo de bits. Estas restricciones pueden ser simples límites de valores. Alternativamente, pueden tomar la forma de restricciones sobre combinaciones aritméticas de valores (por ejemplo, ancho de la imagen multiplicado por la altura de la imagen multiplicado por el número de imágenes decodificadas por segundo). Un nivel especificado para un nivel inferior es más limitado que un nivel especificado para un nivel superior.

En algunos ejemplos, una capa y un nivel pueden imponer limitaciones al decodificador (por ejemplo, el decodificador de video 30) que procesa la carga y las capacidades de memoria del decodificador. La información de nivel puede establecer restricciones en el procesamiento de datos. Por ejemplo, la información de nivel puede incluir la frecuencia de muestreo máxima, el tamaño máximo de la imagen, la tasa de bits máxima y la relación de compresión mínima, entre otras posibilidades. La información de nivel también puede incluir capacidades del DPB y de la memoria intermedia de imagen codificada (CPB). El CPB contiene datos comprimidos antes de su decodificación para fines de gestión del flujo de datos. El DPB contiene datos de imagen decodificados.

Las capas y niveles de codificación de video de alta eficiencia son restricciones que definen un flujo de bits de codificación de video de alta eficiencia (HEVC) en términos de tasa de bits máxima, frecuencia de muestreo de luma máxima, tamaño de imagen de luma máximo, relación de compresión mínima, número máximo de segmentos permitidos y número máximo de mosaicos permitidos. Las capas más bajas se encuentran más restringidas que las capas más altas y los niveles más bajos están más restringidos que los niveles más altos.

El estándar HEVC define dos capas: Principal y Alta. La capa principal es una capa más baja que la capa alta. Las capas se implementan para manejar aplicaciones que difieren en términos de su tasa de bits máxima. La capa principal puede acomodar muchas aplicaciones comunes con tasas de bits promedio, mientras que la capa alta puede acomodar aplicaciones más exigentes.

El estándar HEVC define trece niveles. Un nivel es un conjunto de restricciones para un flujo de bits. Para los niveles inferiores al nivel 4, solo se permite la capa Principal. Un decodificador de video 30 que se ajusta a una capa/nivel dado es capaz de decodificar todos los flujos de bits que se codifican para esa capa/nivel y para todas las capas/niveles inferiores.

Como se describe en la presente memoria, el codificador de video (por ejemplo, el codificador de video 20 o el decodificador de video 30 (Figura 1)) puede transmitir y recibir estructuras de sintaxis de perfil, capa y nivel a través del enlace 16. El codificador de video (por ejemplo, el sistema 10) también puede referirse a una de las estructuras de sintaxis de perfil, capa y nivel para cada uno de una pluralidad de conjuntos de capas de salida y decodificar datos de video de uno o más de los conjuntos de capas de salida en base a la información de la estructura de sintaxis de perfil, capa y nivel a la que se hace referencia para el OLS. Por ejemplo, el codificador de video 20 puede hacer referencia a una de las estructuras de sintaxis de perfil, capa y nivel para cada uno de una pluralidad de conjuntos de capas de salida y codificar datos de video de uno o más conjuntos de capas de salida en base a la información de estructuras de sintaxis de perfil, capa y nivel a la que se hace referencia para el conjunto de capas de salida. El decodificador de video 30 también puede referirse a una de las estructuras de sintaxis de perfil, capa y nivel para cada uno de una pluralidad de conjuntos de capas de salida y decodificar datos de video de uno de los conjuntos de capas de salida en base a la información de las estructuras de sintaxis de perfil, capa y nivel a la que se hace referencia para el conjunto de capas de salida.

La información de PTL puede contenerse dentro de los elementos de sintaxis `profile_tier_level()` y transmitirse a través de conjuntos de parámetros (por ejemplo, SPS, VPS). La estructura de sintaxis `profile_tier_level()` puede proporcionar el perfil, capa y nivel al que se ajusta un OLS y puede incluirse en el VPS activo para el OLS. En algunos ejemplos, la información de PTL se incluye en la extensión VPS. Cuando la estructura de sintaxis `profile_tier_level()` se incluye en una estructura de sintaxis `vps_extension()`, el elemento de sintaxis `profile_tier_level_idx[i]` de la estructura de sintaxis `vps_extension()` especifica qué estructura de sintaxis `profile_tier_level()` se aplica al i-ésimo OLS, donde "i" es una capa entera. Cuando la sintaxis `num_add_layer_sets` es mayor que 0, e i está en el rango de `FirstAddLayerSetIdx` a `LastAddLayerSetIdx`, inclusive, la estructura de sintaxis `profile_tier_level()` identificada por `profile_tier_level_idx[i]` se aplica a la salida de una capa no base con la variable de entrada `lidx` establecida igual a `OlIdxToLsIdx[i]`. Cuando la estructura de sintaxis `profile_tier_level()` se incluye en un VPS, pero no en una estructura de sintaxis `vps_extension()`, se aplica al 0<sup>ésimo</sup> OLS. Cuando la estructura de sintaxis `profile_tier_level()` se incluye en un SPS activo para la capa base, se aplica al 0<sup>ésimo</sup> OLS. Cuando la estructura de sintaxis `profile_tier_level()` se incluye en un SPS activo para una capa no base independiente con `nuh_layer_id` igual a `layerId`, se aplica al 0<sup>ésimo</sup> OLS en un flujo de bits de salida del procedimiento de extracción del subárbol de la capa no base con un parámetro de entrada `lidx` tal que `AssignedBaseLayerId[lidx]` es igual a `layerId`.

La Figura 4 es un diagrama de flujo de señales que muestra un ejemplo de flujo de señales entre un codificador de video y un decodificador de video. Un flujo de señal 400 es representativo del codificador de video 20 que recibe datos de video 405 de, por ejemplo, la fuente de video 18. Los datos de video 405 pueden incluir información de video sin procesar que el codificador de video 20 puede codificar en una capa base (BL) o capa 0, y una o más

capas no base, o capas de mejora (EL). En algunos ejemplos, el codificador de video 20 puede codificar los datos de video 405 como datos de video de la capa base 410 (por ejemplo, capa 0) y datos de video de múltiples capas 415 (por ejemplo, capas 0-2). El codificador de video 20 puede recibir los datos de video 405 para codificarlos y transmitirlos al decodificador de video 30.

En algunas realizaciones, el codificador de video 20 puede señalar un VPS 420 con un `nuh_layer_id` igual a cero (0) que incluye una información de perfil, capa y nivel para una capa base de datos de video. El `nuh_layer_id` puede usarse para indicar una capa correspondiente de datos de video. En consecuencia, el VPS 420 puede denominarse el "VPS activo", hasta que se reemplace por un VPS próximo o siguiente. Una capa puede ser un conjunto de unidades VCL NAL que tienen un valor particular de `nuh_layer_id` y las unidades asociadas no VCL NAL. El `nuh_layer_id` puede ser igual a 0 en un flujo de bits que contiene solo una capa, como, por ejemplo, la capa base o el 0<sup>ésimo</sup> OLS. El `nuh_layer_id` también puede ser igual a 0 en todas las unidades NAL de la capa base. Otros valores de `nuh_layer_id` pueden aparecer en flujos de bits que contienen múltiples capas como, por ejemplo, los datos de video de múltiples capas 415.

El codificador de video 20 puede transmitir el VPS 420 con un `nuh_layer_id` igual a 0. En tal ejemplo, el VPS 420 puede incluir la estructura de sintaxis de perfil, capa y nivel para los datos de video de la capa base 410, o un 0<sup>ésimo</sup> OLS 426 porque la capa base es el único contenido del flujo de bits y, por lo tanto, la única capa de salida. El decodificador de video 30 puede recibir el VPS 420 con el `nuh_layer_id` igual a 0, en el que el VPS incluye una estructura de sintaxis de perfil, capa y nivel para los datos de video de la capa base 410 y utilizó la información de PTL para decodificar el 0<sup>ésimo</sup> OLS 426.

El codificador de video 20 también puede transmitir un SPS 422 con un `nuh_layer_id` igual a 0, al hacer referencia al VPS 420 e indicar información de PTL para los datos de video de la capa base 410. El codificador de video 20 puede transmitir además unidades VCL NAL 424 que incluyen los datos de video codificados 405. En algunas realizaciones, las unidades VCL NAL 424 pueden ser representativas de todo el flujo de bits que tiene solo la capa base (capa 0) como salida, por ejemplo, el 0<sup>ésimo</sup> OLS 426. El decodificador de video 30 puede entonces hacer referencia al VPS 420 para decodificar el 0<sup>ésimo</sup> OLS 426. En algunas realizaciones, el VPS 420 y el SPS 422 pueden denominarse SPS "activo" o VPS activo si la información de PTL que contiene cada uno se referencia actualmente para decodificar los datos de video 405 que se reciben en el decodificador de video 30. En algunos ejemplos, un SPS (por ejemplo, el SPS 422) contiene una estructura de sintaxis de PTL o una cero, mientras que un VPS puede contener múltiples estructuras de sintaxis de PTL, por ejemplo, una para cada capa.

En tal realización, el 0<sup>ésimo</sup> OLS especificado por el VPS activo se refiere al OLS que contiene solo los datos de video de la capa base 410 (la capa con `nuh_layer_id` igual a 0), que también es la única capa de salida. En consecuencia, cuando todo el flujo de bits contiene solo los datos de video de la capa base 410, la información de capa y nivel del 0<sup>ésimo</sup> OLS 426 se señala en la estructura de sintaxis `profile_tier_level ()` del SPS 420 activo, así como en la primera estructura de sintaxis `profile_tier_level ()` en el VPS 422 activo.

En algunas realizaciones, puede usarse un procedimiento similar para codificar datos 415 de video multicapa. El codificador de video 20 puede señalar un VPS 430. El VPS 430 puede incluir información de PTL para cada OLS en el flujo de bits. Por ejemplo, la información de PTL puede incluirse para un 0<sup>ésimo</sup> OLS 450, un 1-ésimo OLS o un 2-ésimo OLS. El codificador de video 20 también puede señalar un SPS 432 que incluye información de PTL para todo el flujo de bits (por ejemplo, los datos de video 415). El codificador de video 20 puede luego transmitir unidades VCL NAL 438 que tienen uno o más OLS identificados en el VPS 430. Debido a que el VPS 430 contiene información de PTL para cada OLS en el flujo de bits, el decodificador de video 30 puede hacer referencia al VPS 430 y recibir información de PTL para un 0<sup>ésimo</sup> OLS 450 cuando decodifica el flujo de bits.

En algunas realizaciones, la `vps_extension ()` de la estructura de sintaxis, de por ejemplo el VPS 430, puede incluir información de sintaxis de PTL para el 0<sup>ésimo</sup> OLS. En consecuencia, la información de PTL para el 0<sup>ésimo</sup> OLS puede implementarse al comienzo de la extensión VPS, como se muestra a continuación. Esto puede mejorar la eficiencia de la extensión VPS al proporcionar la información de PTL necesaria en el nivel apropiado y disminuir la señalización adicional para recibir dicha información de PTL cuando no se incluye. La Tabla 1 muestra los elementos de sintaxis que pueden incluirse en la extensión VPS cuando existen tales condiciones.

Algunas adiciones a la estructura de sintaxis de `vps_extension ()` pueden incluir los siguientes elementos de sintaxis, como se enumeran en la Tabla 1.

TABLA 1

Porción de ejemplo de la sintaxis de VPS	
vps_extension () {	Descriptor
...	
NumOutputLayerSets = num_add_olss + NumLayerSets	
para (¿i = vps_base_laver_internal_flag? 0: 1; i < NumOutputLayerSets; i++) {	
si (i >= NumLayerSets)	
layer_set_idx_for_ols_minus1[ i ]	u(v)
si (i > vps_num_layer_sets_minus1    (defaultOutputLayerIdc == 2 && i > 0))	
para (j = 0; j < NumLayersInIdList [ OlsIdxToLsIdx [ i ] ]; j++)	
output_layer_flag [ i ] [ j ]	u (1)
profile_level_tier_idx [ i ]	u(v)
si (NumOutputLayersInOutputLayerSet [ i ] == 1	
alt_output_layer_flag [ i ]	u (1)
}	
...	
}	

Como se muestra en la Tabla 1, la sintaxis vps\_extension () puede incorporar el término "para (¿i = vps\_base\_laver\_internal\_flag? 0: 1; i < NumOutputLayerSets; i++)" y "si (i > vps\_num\_layer\_sets\_minus1 || (defaultOutputLayerIdc == 2 && i > 0))" para incluir información de PTL para el 0<sup>ésimo</sup> OLS. Esto puede garantizar que, aunque el flujo de bits contenga más que la capa base (por ejemplo, datos de video multicapa 415) y más de un OLS (por ejemplo, el 0<sup>ésimo</sup>, 1-ésimo y 2-ésimo OLS), la información de capa y nivel siempre se indica para el 0<sup>ésimo</sup> OLS.

La Figura 5 es un diagrama de flujo de un procedimiento para codificar un flujo de bits. En el bloque 510, genere uno o más elementos de sintaxis para su inclusión en un VPS, lo que indica una pluralidad de conjuntos de capas de salida (OLS) dentro de un flujo de bits. Los elementos de sintaxis también pueden incluirse en la parte de extensión del VPS (por ejemplo, el VPS 430). Cada OLS puede tener una o más capas de la pluralidad de capas. Los elementos de sintaxis pueden indicar información de perfil, capa y nivel (PTL) para cada OLS de la pluralidad de OLS contenidas en el flujo de bits.

En el bloque 520, el codificador de video 20 puede codificar la pluralidad de capas que incluyen los elementos de sintaxis dentro del VPS (por ejemplo, el VS 430). De esta manera, el procedimiento 500 puede usarse para señalar información de perfil, capa y nivel para cada OLS de un flujo de bits, incluida la información de PTL para el 0<sup>ésimo</sup> OLS, en la extensión VPS.

La Figura 6 es un diagrama de flujo de un procedimiento para decodificar un flujo de bits. Un procedimiento 600 comienza en el bloque 610 cuando el decodificador de video 30 recibe un flujo de bits que tiene una pluralidad de capas de datos de video. El flujo de bits puede incluir los datos de video de múltiples capas 415 (Figura 4). La pluralidad de capas de datos de video también puede incluir una capa base y un VPS (por ejemplo, el VPS 430). El VPS puede indicar la presencia de una pluralidad de OLS e información de perfil, capa y nivel (PTL) para cada OLS de la pluralidad de OLS.

En el bloque 620, el decodificador de video 30 puede seleccionar uno de los OLS como OLS objetivo para la decodificación. Luego, en el bloque 630, el decodificador de video 30 puede decodificar la OLS objetivo mediante el uso de la información de PTL señalada en el VPS activo (por ejemplo, el VPS 430). En consecuencia, el procedimiento 600 puede proporcionar un procedimiento mediante el cual el decodificador de video 30 puede usar información PTL óptima para cada OLS contenida en el flujo de bits (por ejemplo, el 0<sup>ésimo</sup> OLS, 1-ésimo OLS y 2-ésimo OLS).

Los diversos bloques lógicos, módulos, circuitos y etapas de algoritmo ilustrativos descritos en relación con las realizaciones divulgadas en la presente memoria pueden implementarse como hardware electrónico, software informático o combinaciones de ambos. Para ilustrar claramente esta intercambiabilidad de hardware y software, diversos componentes, bloques, módulos, circuitos y etapas ilustrativas se describieron anteriormente de manera general en términos de su funcionalidad. Si tal funcionalidad se implementa como hardware o software depende de la solicitud particular y de las restricciones de diseño impuestas en el sistema general. Los artesanos expertos pueden implementar la funcionalidad descrita de diversos modos para cada solicitud particular, pero tales decisiones de implementación no deben interpretarse como que provocan una desviación del ámbito de la presente divulgación.

Los procedimientos descritos en la presente memoria pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Dichos procedimientos pueden implementarse en cualquiera de una variedad de dispositivos tales como ordenadores de uso general, dispositivos de comunicación inalámbricos o dispositivos de circuitos integrados que tienen múltiples usos, incluida la aplicación en dispositivos de comunicación inalámbricos y

otros dispositivos. Todas las características descritas como módulos o componentes pueden implementarse juntas en un dispositivo lógico integrado o por separado como dispositivos lógicos discretos pero interoperables. Si se implementan en software, los procedimientos pueden realizarse, al menos en parte, mediante un medio de almacenamiento de datos legible por ordenador que comprende un código de programa que incluye instrucciones que, cuando se ejecutan, realizan uno o más de los procedimientos descritos anteriormente. El medio de almacenamiento de datos legible por ordenador puede formar parte de un producto de programa informático, que puede incluir materiales de embalaje. El medio legible por ordenador puede comprender memoria o medios de almacenamiento de datos, como memoria de acceso aleatorio (RAM), como memoria de acceso aleatorio dinámico síncrono (SDRAM), memoria de solo lectura (ROM), memoria de acceso aleatorio no volátil (NVRAM), memoria de sólo lectura programable y borrable eléctricamente (EEPROM), memoria FLASH, medios de almacenamiento de datos magnéticos u ópticos, y similares. Los procedimientos, adicional o alternativamente, pueden realizarse, al menos en parte, por un medio de comunicación legible por ordenador que transporta o comunica código de programa en forma de instrucciones o estructuras de datos y al que puede accederse, leerse y/o ejecutarse por un ordenador, como señales u ondas propagadas.

El código del programa puede ejecutarse por un procesador, que puede incluir uno o más procesadores, tal como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), matrices lógicas programables en campo (FPGA), u otro circuito lógico integrado o discreto equivalente. Dicho procesador puede configurarse para realizar cualquiera de los procedimientos descritos en esta divulgación. Un procesador de propósito general puede ser un microprocesador; pero como alternativa, el procesador puede ser cualquier procesador, controlador, microcontrolador o máquina de estado convencional. Un procesador puede implementarse además como una combinación de dispositivos informáticos, por ejemplo, una combinación de un DSP y un microprocesador, una pluralidad de microprocesadores, uno o más microprocesadores junto con un núcleo de DSP o cualquier otra de tal configuración. En consecuencia, el término "procesador", como se usa en la presente memoria puede referirse a cualquiera de las estructuras anteriores, cualquier combinación de las estructuras anteriores, o cualquier otra estructura o aparato adecuado para la implementación de los procedimientos descritos en la presente memoria. Además, en algunos aspectos, la funcionalidad descrita en la presente memoria puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y decodificación, o incorporarse en un codificador-decodificador de video combinado (CODEC).

Los procedimientos de codificación discutidos en la presente memoria pueden ser una realización en un ejemplo de sistema de codificación y decodificación de video. Un sistema incluye un dispositivo fuente que proporciona datos de video codificados para decodificarse en un momento posterior mediante un dispositivo de destino. En particular, el dispositivo fuente proporciona los datos de video al dispositivo de destino a través de un medio legible por ordenador. El dispositivo fuente y el dispositivo de destino pueden comprender cualquiera de una amplia gama de dispositivos, que incluyen ordenadores de escritorio, ordenadores portátiles (es decir, laptop), tabletas, módulos de conexión, teléfonos móviles tal como los denominados teléfonos "inteligentes", las denominadas almohadillas "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de videos o similares. En algunos casos, el dispositivo fuente y el dispositivo de destino pueden equiparse para la comunicación inalámbrica.

Aunque las realizaciones de la divulgación se describen anteriormente para una realización particular, son posibles muchas variaciones de la divulgación. Por ejemplo, el número de varios componentes puede aumentarse o disminuirse, los módulos y las etapas que determinan un voltaje de suministro pueden modificarse para determinar una frecuencia, otro parámetro del sistema o una combinación de parámetros. Además, las características de las diversas realizaciones pueden combinarse en combinaciones que difieren de las descritas anteriormente.

Los expertos apreciarán que los diversos bloques y módulos ilustrativos descritos en relación con la realización divulgada en la presente memoria pueden implementarse de diversas formas. Algunos bloques y módulos se describieron anteriormente en general en términos de su funcionalidad. La forma en que se implementa dicha funcionalidad depende de las limitaciones de diseño impuestas a un sistema general. Los expertos pueden implementar la funcionalidad descrita de diferentes maneras para cada aplicación en particular, pero tales decisiones de implementación no deben interpretarse como que provocan una desviación del ámbito de la divulgación. Además, la agrupación de funciones dentro de un módulo, bloque o etapa es para facilitar la descripción. Las funciones o etapas específicas pueden moverse de un módulo o bloque o distribuirse a través de módulos o bloques sin apartarse de la divulgación.



## REIVINDICACIONES

1. Un procedimiento para codificar un flujo de bits de acuerdo con HEVC escalable (codificación de video de alta eficiencia), SHVC, una extensión 3D de HEVC o una extensión de vista múltiple de HEVC, MV-HEVC, teniendo el flujo de bits una pluralidad de capas de datos de video, incluyendo la pluralidad de capas de datos de video una capa base, comprendiendo el procedimiento:

generar (510) uno o más elementos de sintaxis que indican una pluralidad de conjuntos de capas de salida (OLS) dentro del flujo de bits, teniendo cada OLS una o más capas de la pluralidad de capas, uno de la pluralidad de conjuntos de capas de salida que es un 0<sup>ésimo</sup> OLS que incluye solo la capa base, los elementos de sintaxis que indican además información de perfil, capa y nivel (PTL) para cada OLS de la pluralidad de OLS; y codificar (520) el uno o más elementos de sintaxis que indican la pluralidad de OLS en un conjunto de parámetros de video (VPS), en el que la información de PTL para cada OLS, que incluye el 0<sup>ésimo</sup> OLS, se incluye en una parte de extensión de VPS del VPS.

2. Un dispositivo para codificar un flujo de bits de acuerdo con HEVC escalable (codificación de video de alta eficiencia), SHVC, una extensión 3D de HEVC o una extensión de vista múltiple de HEVC, MV-HEVC, teniendo el flujo de bits una pluralidad de capas de datos de video, incluyendo la pluralidad de capas de datos de video una capa base, comprendiendo el dispositivo:

una memoria configurada para almacenar la pluralidad de capas de datos de video; y al menos, un procesador acoplado operativamente a la memoria y configurado para:

generar uno o más elementos de sintaxis que indican una pluralidad de conjuntos de capas de salida (OLS) dentro del flujo de bits, teniendo cada OLS una o más capas de la pluralidad de capas, uno de la pluralidad de conjuntos de capas de salida que es un 0<sup>ésimo</sup> OLS que incluye solo la capa base, los elementos de sintaxis que indican además información de perfil, capa y nivel (PTL) para cada OLS de la pluralidad de OLS; y codificar el uno o más elementos de sintaxis que indican la pluralidad de OLS en un conjunto de parámetros de video (VPS), en el que la información de PTL para cada OLS, que incluye el 0<sup>ésimo</sup> OLS, se incluye en una parte de extensión de VPS del VPS.

3. Un procedimiento para decodificar datos de video de acuerdo con HEVC escalable (codificación de video de alta eficiencia), SHVC, una extensión 3D de HEVC o una extensión de vista múltiple de HEVC, MV-HEVC, comprendiendo el procedimiento:

recibir (610) un flujo de bits que tiene una pluralidad de capas de datos de video, incluyendo la pluralidad de capas de datos de video una capa base, comprendiendo el flujo de bits además un conjunto de parámetros de video (VPS), incluyendo el VPS uno o más elementos de sintaxis que indican una pluralidad de conjuntos de capas de salida (OLS) e información de perfil, capa y nivel (PTL) para cada OLS de la pluralidad de OLS, uno de la pluralidad de conjuntos de capas de salida que es un 0<sup>ésimo</sup> OLS que incluye solo la capa base; seleccionar (620) el 0<sup>ésimo</sup> OLS de la pluralidad de capas como un OLS objetivo, y decodificar (630) el OLS objetivo en base a la información de PTL señalada en el VPS, en el que la información de PTL para cada OLS que incluye el 0<sup>ésimo</sup> OLS se incluye en una extensión VPS del VPS.

4. Un dispositivo para decodificar datos de video de acuerdo con HEVC escalable (codificación de video de alta eficiencia), SHVC, una extensión 3D de HEVC o una extensión de vista múltiple de HEVC, MV-HEVC, comprendiendo el dispositivo:

una memoria configurada para almacenar un flujo de bits que tiene una pluralidad de capas de datos de video, incluyendo la pluralidad de capas de datos de video una capa base, comprendiendo el flujo de bits además un conjunto de parámetros de video (VPS), incluyendo el VPS uno o más elementos de sintaxis que indican una pluralidad de conjuntos de capas de salida (OLS) e información de perfil, capa y nivel (PTL) para cada OLS de la pluralidad de OLS, uno de la pluralidad de conjuntos de capas de salida que es un 0<sup>ésimo</sup> OLS que incluye solo la capa base; y

al menos, un procesador acoplado operativamente a la memoria y configurado para seleccionar el 0<sup>ésimo</sup> OLS de la pluralidad de capas como un OLS objetivo, y decodificar el OLS objetivo en base a la información de PTL señalada en el VPS, en el que la información de PTL para cada OLS que incluye el 0<sup>ésimo</sup> OLS se incluye en una extensión VPS del VPS.

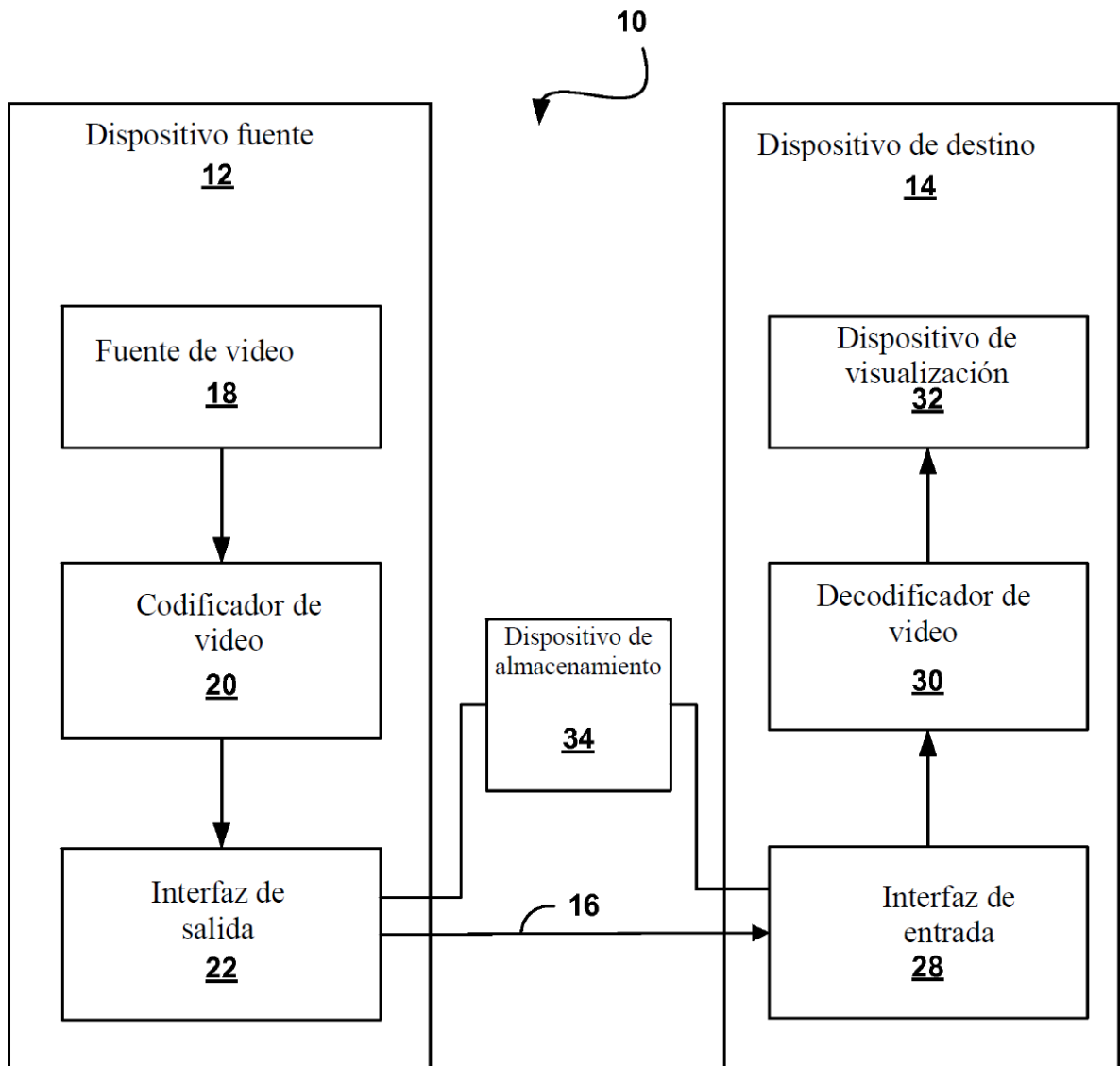


Figura 1

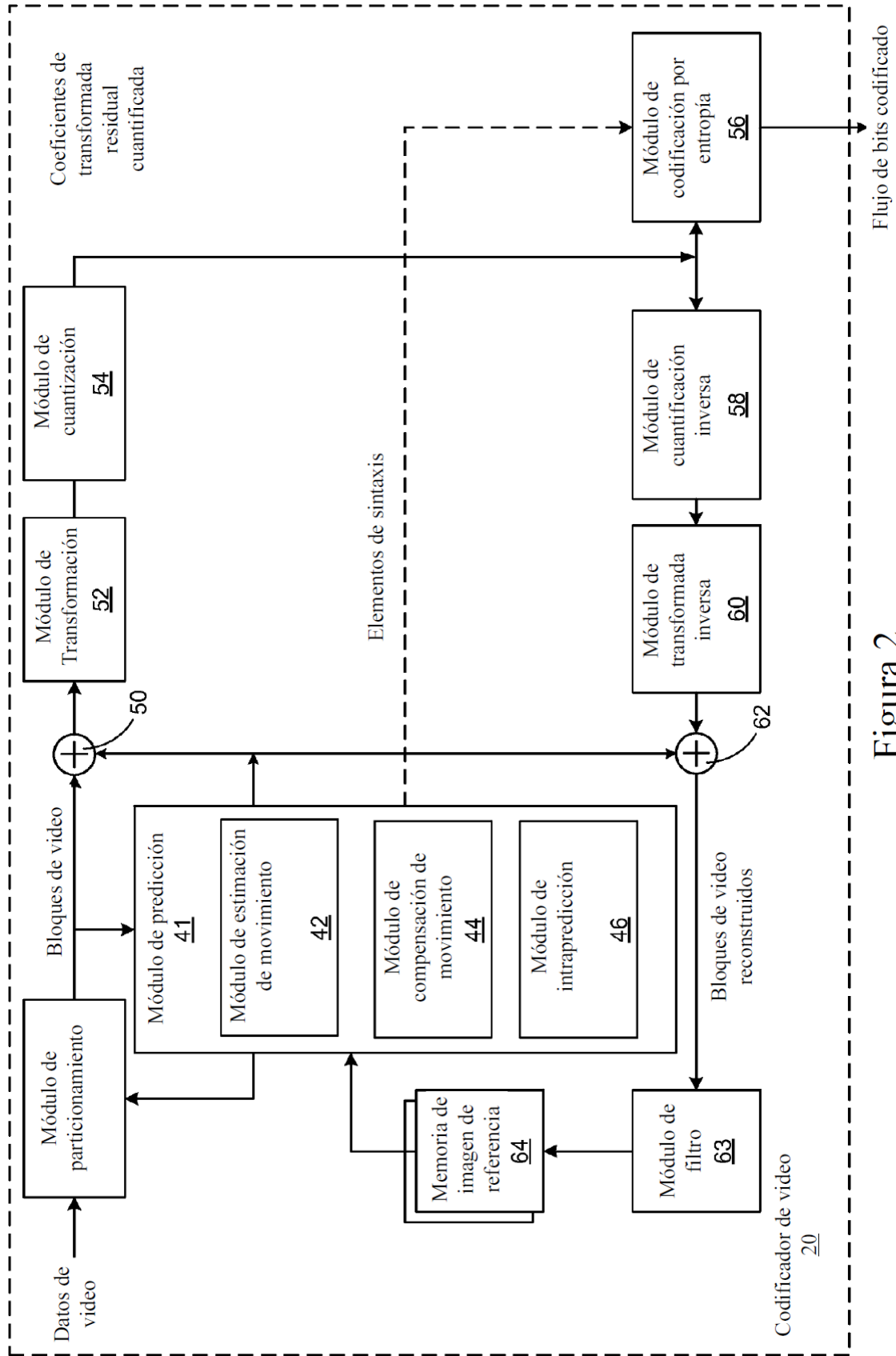


Figura 2

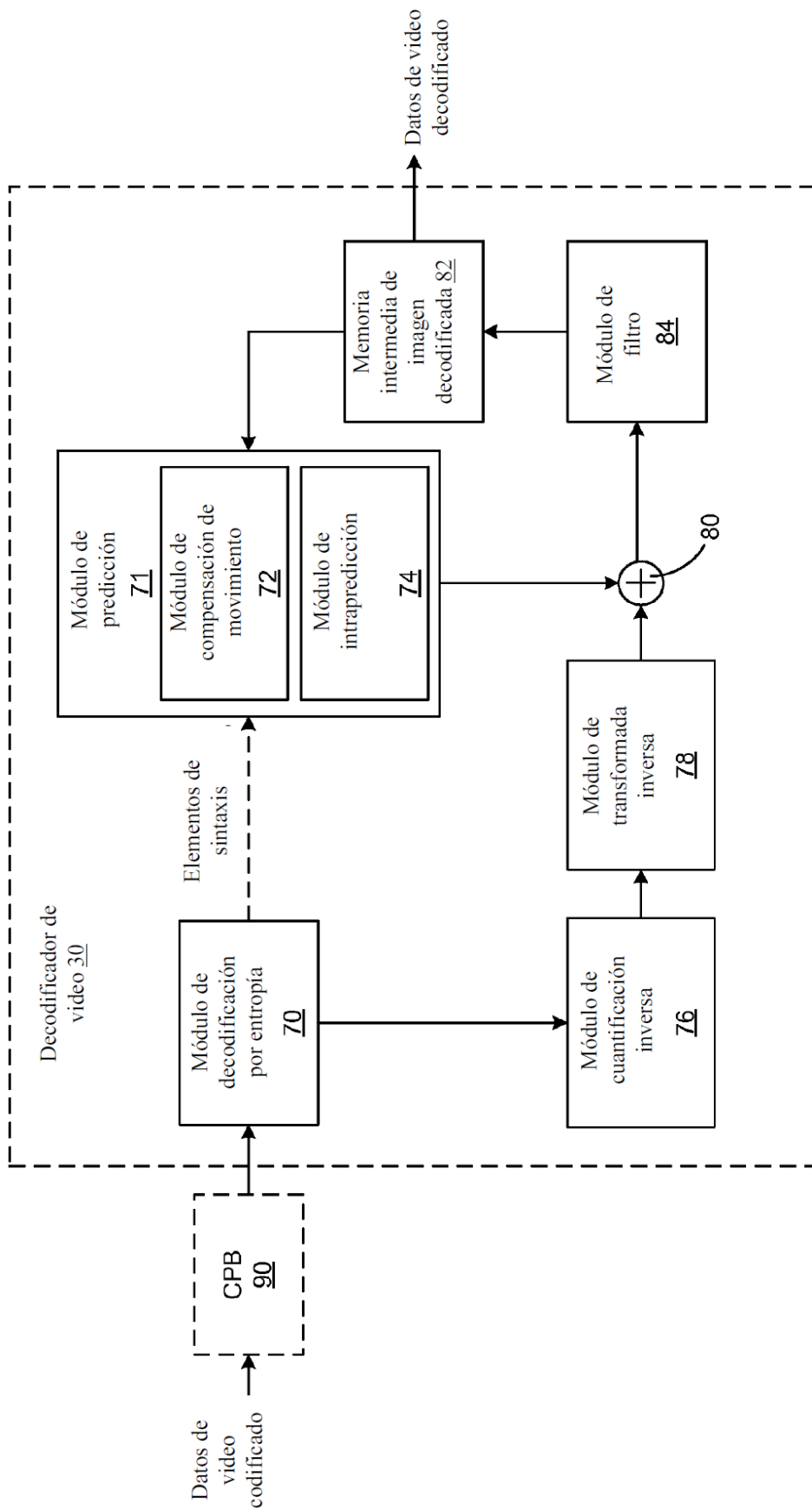


Figura 3

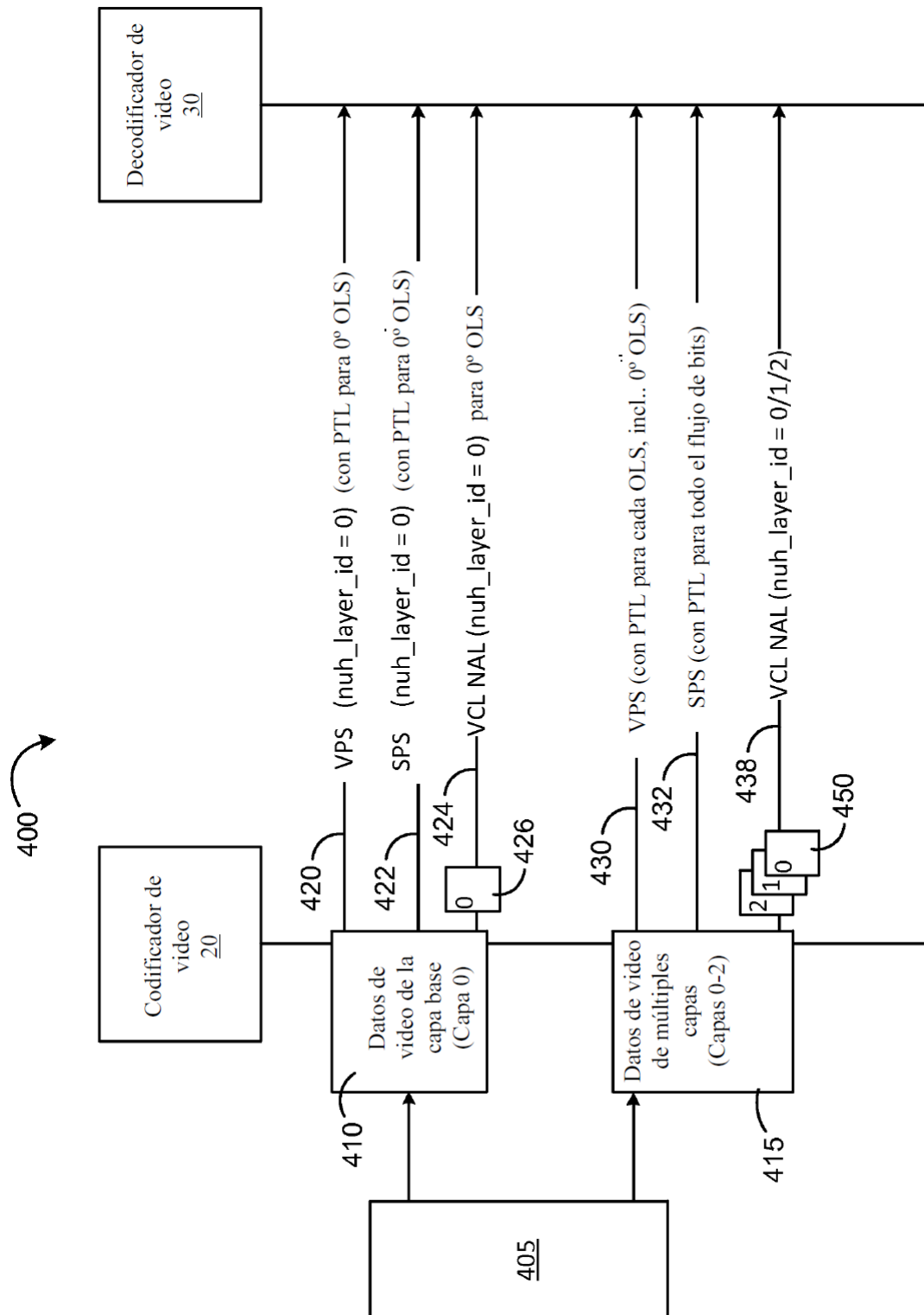


Figura 4

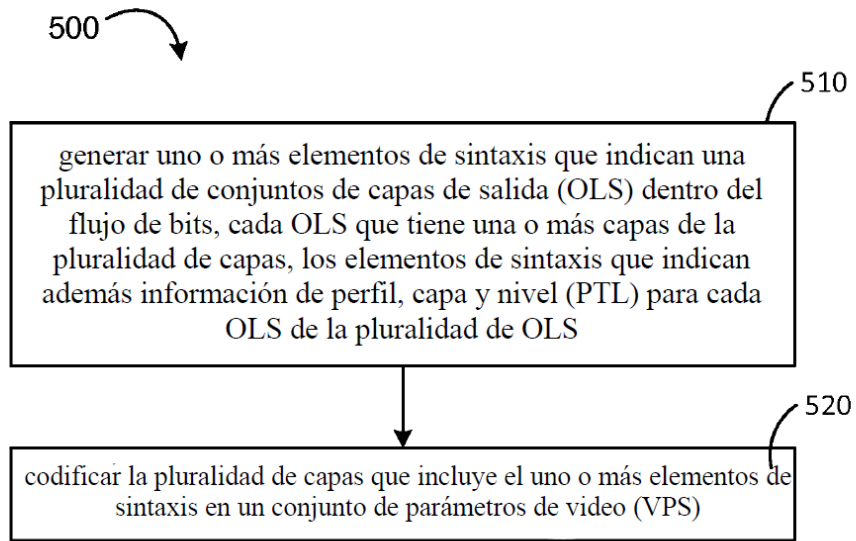


Figura 5

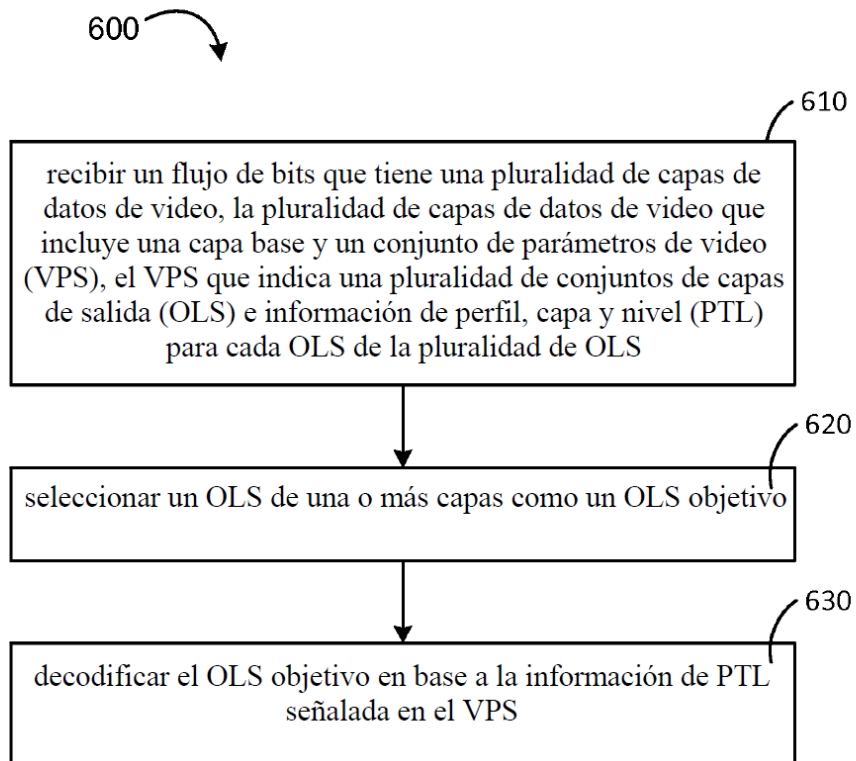


Figura 6