

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 663 692**

51 Int. Cl.:

H04N 19/70 (2014.01)

H04N 19/46 (2014.01)

H04N 19/30 (2014.01)

H04N 19/463 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **20.09.2013 PCT/US2013/060940**

87 Fecha y número de publicación internacional: **17.04.2014 WO14058600**

96 Fecha de presentación y número de la solicitud europea: **20.09.2013 E 13770819 (4)**

97 Fecha y número de publicación de la concesión europea: **17.01.2018 EP 2904782**

54 Título: **Aplicabilidad de sub-flujo de bits a mensajes SEI de anidado en codificación de vídeo**

30 Prioridad:

08.10.2012 US 201261711098 P
30.07.2013 US 201313954758

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
16.04.2018

73 Titular/es:

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

72 Inventor/es:

WANG, YE-KUI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 663 692 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

Aplicabilidad de sub-flujo de bits a mensajes SEI de anidado en codificación de vídeo

5 **CAMPO TÉCNICO**

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

ANTECEDENTES

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

25 [0003] Las técnicas de compresión de vídeo llevan a cabo la predicción espacial (intraimagen) y/o la predicción
 temporal (interimagen) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la
 30 codificación de vídeo basada en bloques, un fragmento de vídeo (es decir, una trama de vídeo o una parte de una
 trama de vídeo) puede dividirse en bloques de vídeo. Los bloques de vídeo en un fragmento intra-codificado (I) de
 una imagen se codifican mediante la predicción espacial con respecto a las muestras de referencia en bloques
 contiguos en la misma imagen. Los bloques de vídeo en un fragmento inter-codificado (P o B) de una imagen
 pueden usar la predicción espacial con respecto a las muestras de referencia en bloques contiguos en la misma
 imagen, o la predicción temporal con respecto a las muestras de referencia en otras imágenes de referencia. Las
 imágenes pueden denominarse fotogramas, y las imágenes de referencia pueden denominarse fotogramas de
 referencia.

35 [0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a
 codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque
 predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de
 40 muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el
 bloque codificado y el bloque predictivo. Un bloque intra-codificado se codifica de acuerdo con un modo de intra-
 codificación y a los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde
 el dominio del píxel a un dominio de transformada, dando como resultado coeficientes residuales, los cuales pueden
 cuantificarse posteriormente. Los coeficientes cuantificados, inicialmente dispuestos en una matriz bidimensional,
 pueden escanearse con el fin de producir un vector unidimensional de coeficientes, y puede aplicarse la codificación
 por entropía para lograr aún más compresión.

45 [0005] Un flujo de bits de codificación de múltiples visualizaciones puede generarse codificando visualizaciones, por
 ejemplo, desde múltiples perspectivas. Se han desarrollado algunas normas de vídeo tridimensionales (3D) que
 hacen uso de aspectos de codificación de múltiples visualizaciones. Por ejemplo, diferentes visualizaciones pueden
 50 transmitir visualizaciones de ojo izquierdo y derecho para soportar vídeo 3D. De forma alternativa, algunos procesos
 de codificación de vídeo 3D pueden aplicar la denominada codificación de múltiples visualizaciones más
 profundidad. En la codificación de múltiples visualizaciones más profundidad, un flujo de bits de vídeo 3D puede
 contener no solo componentes de visualización de textura, sino también componentes de visualización de
 profundidad. Por ejemplo, cada visualización puede comprender un componente de visualización de textura y un
 componente de visualización de profundidad.

55 [0006] El documento "AHG9: Puntos de funcionamiento en VPS y SEI de anidado", 11. Reunión de JCT-VC; 102
 Reunión de MPEG; 10-10-2012-19-10-2015; Shanghái, de Hannuksela (Nokia) MM et al propusieron un mensaje
 SEI de anidado escalable para indicar que los mensajes SEI de anidado pertenecen a un rango de sub-capas
 60 temporales e indican un período de almacenamiento en memoria intermedia y sincronización de imágenes para
 cualquier subconjunto de flujo de bits utilizando un conjunto de identificadores de capa.

SUMARIO

65 [0007] En general, esta divulgación describe la indicación de parámetros de decodificador de referencia hipotético
 (HRD) y el anidado de mensajes de información suplementaria de mejora (SEI) en la codificación de vídeo. Más
 específicamente, un codificador de vídeo puede incluir, en un mensaje SEI de anidado escalable encapsulado por
 una unidad de capa de abstracción de red SEI (NAL), un elemento sintáctico que indica si un mensaje SEI de

anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto. El elemento sintáctico no está en ninguno de los mensajes SEI de anidado. El sub-flujo de bits por defecto es una representación del punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL. Además, un dispositivo puede determinar, basándose al menos en parte en el elemento sintáctico en el mensaje SEI de anidado escalable, si el mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable al sub-flujo de bits por defecto. Cuando el mensaje SEI de anidado es aplicable al sub-flujo de bits por defecto, el dispositivo puede usar el mensaje SEI de anidado en un funcionamiento en el sub-flujo de bits por defecto.

[0008] En un ejemplo, esta divulgación describe un procedimiento para procesar datos de vídeo. El procedimiento comprende determinar, basándose al menos en parte en un elemento sintáctico en un mensaje SEI de anidado escalable encapsulado por una unidad SEI NAL, si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto de un flujo de bits de vídeo codificado. El elemento sintáctico no está en ninguno de los mensajes SEI de anidado. El sub-flujo de bits por defecto es una representación del punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL. Además, el procedimiento comprende, cuando el mensaje SEI de anidado es aplicable al sub-flujo de bits por defecto, usar el mensaje SEI de anidado en un funcionamiento en el sub-flujo de bits por defecto.

[0009] En otro ejemplo, esta divulgación describe un dispositivo que comprende uno o más procesadores configurados para determinar, basándose al menos en parte en un elemento sintáctico en un mensaje SEI de anidado escalable encapsulado por una unidad SEI NAL, si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto de un flujo de bits de vídeo codificado. El elemento sintáctico no está en ninguno de los mensajes SEI de anidado. El sub-flujo de bits por defecto es una representación del punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL. El uno o más procesadores están configurados de manera tal que cuando el mensaje SEI de anidado es aplicable al sub-flujo de bits por defecto, uno o más procesadores usan el mensaje SEI de anidado en un funcionamiento en el sub-flujo de bits por defecto.

[0010] En otro ejemplo, esta divulgación describe un dispositivo que comprende medios para determinar, basándose al menos en parte en un elemento sintáctico en un mensaje SEI de anidado escalable encapsulado por una unidad SEI NAL, si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto de un flujo de bits de vídeo codificado. El elemento sintáctico no está en ninguno de los mensajes SEI de anidado. El sub-flujo de bits por defecto es una representación del punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL. El dispositivo también comprende medios para usar, cuando el mensaje SEI de anidado es aplicable al sub-flujo de bits por defecto, el mensaje SEI de anidado en un funcionamiento en el sub-flujo de bits por defecto.

[0011] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando son ejecutadas por uno o más procesadores de un dispositivo, configuran el dispositivo para determinar, basándose al menos en parte en un elemento sintáctico en un mensaje SEI de anidado escalable encapsulado por una unidad SEI NAL, si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto de un flujo de bits de vídeo codificado. El elemento sintáctico no está en ninguno de los mensajes SEI de anidado. El sub-flujo de bits por defecto es una representación del punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL. Las instrucciones, cuando se ejecutan, configuran el dispositivo de modo que cuando el mensaje SEI de anidado es aplicable al sub-flujo de bits por defecto, el dispositivo usa el mensaje SEI de anidado en una operación en el sub-flujo de bits por defecto.

[0012] En otro ejemplo, esta divulgación describe un procedimiento para codificar datos de vídeo. El procedimiento comprende incluir, en un mensaje SEI de anidado escalable encapsulado por una unidad SEI NAL, un elemento sintáctico que indica si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto de un flujo de bits de vídeo codificado. El elemento sintáctico no está en ninguno de los mensajes SEI de anidado. El sub-flujo de bits por defecto es una representación del punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL. El procedimiento también comprende indicar el mensaje SEI de anidado escalable en el flujo de bits de vídeo codificado.

[0013] En otro ejemplo, esta divulgación describe un dispositivo de codificación de vídeo que comprende uno o más procesadores configurados para incluir, en un mensaje SEI de anidado escalable encapsulado por una unidad SEI

NAL, un elemento sintáctico que indica si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto de un flujo de bits de vídeo codificado. El elemento sintáctico no está en ninguno de los mensajes SEI de anidado. El sub-flujo de bits por defecto es una representación del punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL. El uno o más procesadores también están configurados para indicar el mensaje SEI de anidado escalable en el flujo de bits de vídeo codificado.

[0014] En otro ejemplo, esta divulgación describe un dispositivo de codificación de vídeo que comprende medios para incluir, en un mensaje SEI de anidado escalable encapsulado por una unidad SEI NAL, un elemento sintáctico que indica si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto de un flujo de bits de vídeo codificado. El elemento sintáctico no está en ninguno de los mensajes SEI de anidado. El sub-flujo de bits por defecto es una representación del punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL. El dispositivo de codificación de vídeo también comprende medios para indicar el mensaje SEI de anidado escalable en el flujo de bits de vídeo codificado.

[0015] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando son ejecutadas por un dispositivo de codificación de vídeo, configuran el dispositivo de codificación de vídeo para incluir, en un mensaje SEI de anidado escalable encapsulado por una unidad SEI NAL, un elemento sintáctico que indica si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto de un flujo de bits de vídeo codificado. El elemento sintáctico no está en ninguno de los mensajes SEI de anidado. El sub-flujo de bits por defecto es una representación del punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL. Las instrucciones, cuando se ejecutan, también configuran el dispositivo de codificación de vídeo para indicar el mensaje SEI de anidado escalable en el flujo de bits de vídeo codificado.

[0016] La información de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y la descripción siguiente. Otras características, objetivos y ventajas resultarán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0017]

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

La FIG. 2 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 3 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 4 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un codificador de vídeo de acuerdo con una o más técnicas de esta divulgación.

La FIG. 5 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un dispositivo, de acuerdo con una o más técnicas de esta divulgación.

La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un codificador de vídeo de acuerdo con una o más técnicas de esta divulgación.

La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un dispositivo, de acuerdo con una o más técnicas de esta divulgación.

La FIG. 8 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un codificador de vídeo de acuerdo con una o más técnicas de esta divulgación.

La FIG. 9 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un dispositivo, de acuerdo con una o más técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

[0018] Un codificador de vídeo puede generar un flujo de bits que incluye datos de vídeo codificados. Debido a que el flujo de bits incluye datos de vídeo codificados, el flujo de bits se puede denominar en el presente documento un flujo de bits de vídeo codificado. El flujo de bits puede comprender una serie de unidades de capa de abstracción de red (NAL). Las unidades de NAL pueden incluir unidades de NAL de capa de codificación de vídeo (VCL) y unidades de NAL no de VCL. Las unidades NAL VCL pueden incluir fragmentos codificados de imágenes. Una unidad NAL no VCL puede incluir un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), información de mejora suplementaria (SEI) u otros tipos de datos. Un VPS es una estructura sintáctica que puede contener elementos sintácticos que se aplican a cero o más secuencias de vídeo codificadas completas. Un SPS es una estructura sintáctica que puede contener elementos sintácticos que se aplican a cero o más secuencias de vídeo codificadas completas. Un solo VPS puede ser aplicable a múltiples SPS. Un PPS es una estructura sintáctica que puede contener elementos sintácticos que se aplican a cero o más imágenes codificadas completas. Un solo SPS puede ser aplicable a múltiples PPS.

[0019] Un dispositivo, tal como un dispositivo de red de distribución de contenidos (CDN), un elemento de red consciente de medios (MANE), un codificador de vídeo, o un decodificador de vídeo, puede extraer un sub-flujo de bits del flujo de bits. El dispositivo puede realizar el proceso de extracción de sub-flujo de bits eliminando ciertas unidades NAL del flujo de bits. El sub-flujo de bits resultante incluye las unidades NAL restantes, no eliminadas del flujo de bits. En algunos ejemplos, los datos de vídeo descodificados desde el sub-flujo de bits pueden tener una velocidad de trama menor y/o pueden representar menos visualizaciones que el flujo de bits original.

[0020] Las normas de codificación de vídeo pueden incluir diversas características para soportar el proceso de extracción de sub-flujo de bits. Por ejemplo, los datos de vídeo del flujo de bits se pueden dividir en un conjunto de capas. Para cada una de las capas, los datos en una capa inferior pueden descodificarse sin referencia a los datos en cualquier capa superior. Las unidades NAL solo encapsulan datos de una sola capa. Por lo tanto, las unidades NAL que encapsulan los datos de la capa restante más alta del flujo de bits pueden eliminarse del flujo de bits sin afectar a la descodificabilidad de los datos en las capas restantes del flujo de bits. En la codificación de vídeo escalable (SVC), las capas superiores pueden incluir datos de mejora que mejoran la calidad de las imágenes en capas inferiores (escalabilidad de calidad), amplían el formato espacial de las imágenes en capas inferiores (escalabilidad espacial) o aumentan la velocidad temporal de las imágenes capas inferiores (escalabilidad temporal). En codificación de múltiples visualizaciones (MVC) y codificación de vídeo tridimensional (3DV), las capas superiores pueden incluir visualizaciones adicionales.

[0021] Cada unidad NAL puede incluir una cabecera y una carga útil. Las cabeceras de las unidades NAL pueden incluir elementos sintácticos `nuh_reserved_zero_6bits`. El elemento sintáctico `nuh_reserved_zero_6bits` de una unidad NAL es igual a dos 0 si la unidad NAL se relaciona con una capa base en MVC, codificación 3DV o SVC. Los datos en una capa base de un flujo de bits pueden descodificarse sin referencia a los datos en cualquier otra capa del flujo de bits. Si la unidad NAL no se relaciona con una capa base en MVC, 3DV o SVC, el elemento sintáctico `nuh_reserved_zero_6bits` puede tener otros valores distintos de cero. Específicamente, si una unidad NAL no se relaciona con una capa base en MVC, 3DV o SVC, el elemento sintáctico `nuh_reserved_zero_6bits` de la unidad NAL especifica un identificador de capa que identifica una capa asociada con la unidad NAL.

[0022] Además, algunas imágenes dentro de una capa se pueden descodificar sin referencia a otras imágenes dentro de la misma capa. Por lo tanto, las unidades NAL que encapsulan datos de ciertas imágenes de una capa pueden eliminarse del flujo de bits sin afectar la capacidad de descodificación de otras imágenes en la capa. Por ejemplo, las imágenes con valores de recuento de orden de imagen (POC) par pueden ser descodificables sin referencia a imágenes con valores POC impares. La eliminación de unidades NAL que encapsulan datos de tales imágenes puede reducir la velocidad de tramas del flujo de bits. Un subconjunto de imágenes dentro de una capa que puede descodificarse sin referencia a otras imágenes dentro de la capa se puede denominar en el presente documento una "sub-capa" o una "sub-capa temporal".

[0023] Las unidades NAL pueden incluir elementos sintácticos `nuh_temporal_id_plus1`. El elemento sintáctico `nuh_temporal_id_plus1` de una unidad NAL puede especificar un identificador temporal de la unidad NAL. Si el identificador temporal de una primera unidad NAL es menor que el identificador temporal de una segunda unidad NAL, los datos encapsulados por la primera unidad NAL pueden descodificarse sin referencia a los datos encapsulados por la segunda unidad NAL.

[0024] Cada uno de los puntos de funcionamiento de un flujo de bits está asociado con un conjunto de identificadores de capa (es decir, un conjunto de valores `nuh_reserved_zero_6bits`) y un identificador temporal. El conjunto de identificadores de capa se puede denotar como `OpLayerIdSet` y el identificador temporal se puede denotar como `TemporalID`. Si el identificador de capa de una unidad NAL se encuentra en el conjunto de identificadores de capa de un punto de funcionamiento y el identificador temporal de la unidad NAL es menor o igual que el identificador temporal del punto de funcionamiento, la unidad NAL está asociada con el punto de funcionamiento. Una representación de punto de funcionamiento es un subconjunto de flujo de bits (es decir, un sub-flujo de bits) que está asociado con un punto de funcionamiento. La representación del punto de funcionamiento de un punto de funcionamiento puede incluir cada unidad NAL que está asociada con el punto de funcionamiento. La representación del punto de funcionamiento no incluye las unidades NAL VCL que no están asociadas con el punto

de funcionamiento.

[0025] Una fuente externa pueden especificar un conjunto de identificadores de capa de destino para un punto de funcionamiento. Por ejemplo, un dispositivo de red de entrega de contenido (CDN) puede especificar el conjunto de identificadores de capa de destino. En este ejemplo, el dispositivo CDN puede usar el conjunto de identificadores de capa de destino para identificar un punto de funcionamiento. El dispositivo CDN puede entonces extraer la representación del punto de funcionamiento para el punto de funcionamiento y enviar la representación del punto de funcionamiento, en lugar del flujo de bits original, a un dispositivo cliente. La extracción y envío de la representación del punto de funcionamiento al dispositivo cliente puede reducir la velocidad de transferencia de bits del flujo de bits.

[0026] Por otra parte, las normas de codificación de vídeo especifican modelos de memoria intermedia de vídeo. Un modelo de almacenamiento en memoria intermedia de vídeo también se puede denominar "descodificador de referencia hipotético" o "HRD". El HRD describe cómo se almacenarán en memoria intermedia los datos para la descodificación y cómo se almacenan en memoria intermedia los datos descodificados para la salida. Por ejemplo, el HRD describe el funcionamiento de una memoria intermedia de imágenes codificadas ("CPB") y una memoria intermedia de imágenes descodificadas ("CPB") en un descodificador de vídeo. El CPB es una memoria intermedia de primero en entrar, primero en salir, que contiene unidades de acceso en el orden de descodificación especificado por HRD. El DPB es una memoria intermedia que contiene imágenes descodificadas para referencia, reordenamiento de salida o retardo de salida especificado por el HARD.

[0027] Un codificador de vídeo puede indicar un conjunto de parámetros HRD. Los parámetros HRD controlan varios aspectos del HRD. Los parámetros HRD pueden incluir un retardo inicial de eliminación del CPB, un tamaño de CPB, una velocidad de transferencia de bits, un retardo inicial de salida del DPB y un tamaño de DPB. Estos parámetros HRD pueden codificarse en una estructura sintáctica `hrd_parameters()` especificada en un VPS y/o un SPS. Los parámetros HRD también se pueden especificar en los mensajes SET de período de almacenamiento en memoria intermedia o mensajes SEI de sincronización de imágenes.

[0028] Como se explicó anteriormente, una representación de punto de funcionamiento puede tener una velocidad de tramas y/o velocidad de transferencia de bits diferente que un flujo de bits original. Esto se debe a que la representación del punto de funcionamiento puede no incluir algunas imágenes y/o algunos de los datos del flujo de bits original. Por lo tanto, si el descodificador de vídeo eliminase datos del CPB y/o del DPB a una velocidad particular al procesar el flujo de bits original y si el descodificador de vídeo eliminara datos del CPB y/o el DPB a la misma velocidad cuando procesa una representación del punto de funcionamiento, el descodificador de vídeo puede eliminar demasiada o muy poca información del CPB y/o del DPB. En consecuencia, el codificador de vídeo puede indicar diferentes conjuntos de parámetros HRD para diferentes puntos de funcionamiento. En la norma emergente de codificación de vídeo de alta eficiencia (HEVC), el codificador de vídeo puede indicar conjuntos de parámetros HRD en un VPS o el codificador de vídeo puede indicar conjuntos de parámetros HRD en un SPS.

[0029] El conjunto de parámetros HRD incluye opcionalmente un conjunto de información que es común para todas las sub-capas temporales. Una sub-capa temporal es una capa temporal escalable de un flujo de bits temporal escalable que consta de unidades NAL VCL con un identificador temporal particular y las unidades NAL no VCL asociadas. Además del conjunto de información común, los conjuntos de parámetros HRD pueden incluir un conjunto de elementos sintácticos que son específicos de sub-capas temporales individuales. Debido a que el conjunto de información común es común a múltiples conjuntos de parámetros HRD, puede ser innecesario indicar el conjunto de información común en múltiples conjuntos de parámetros HRD. En algunas propuestas para HEVC, la información común puede estar presente en un conjunto de parámetros HRD cuando el conjunto de parámetros HRD es el primer conjunto de parámetros HRD en un VPS o la información común puede estar presente en un conjunto de parámetros HRD cuando el conjunto de los parámetros HRD está asociado con el primer punto de funcionamiento.

[0030] Sin embargo, cuando hay múltiples conjuntos de parámetros HRD en un VPS, puede ser deseable tener múltiples conjuntos diferentes de información común para los conjuntos de parámetros HRD. Esto puede ser especialmente cierto cuando hay un gran número de estructuras sintácticas de parámetros HRD en un VPS. Por lo tanto, puede ser deseable tener conjuntos de información común en estructuras sintácticas de parámetros HRD distintas de la primera estructura sintáctica de parámetros HRD.

[0031] Las técnicas de esta divulgación proporcionan un diseño para permitir que la información común de las estructuras sintácticas de parámetros HRD sea indicada explícitamente para cualquier estructura sintáctica de parámetros HRD. En otras palabras, las técnicas de esta divulgación pueden permitir que la información, que es común para todas las sub-capas, sea indicada explícitamente para cualquier estructura sintáctica `hrd_parameters()`. Esto puede mejorar la eficiencia de la codificación.

[0032] Así, de acuerdo con una o más técnicas de esta divulgación, un dispositivo, como un descodificador de vídeo u otro dispositivo, puede determinar, basándose al menos en parte en un elemento sintáctico en un VPS que incluye una pluralidad de estructuras sintácticas de parámetros HRD, si una estructura sintáctica de parámetros HRD particular en el VPS incluye un conjunto de parámetros HRD que son comunes a cada sub-capas de un flujo de bits.

El dispositivo puede descodificar el elemento sintáctico del VPS. Una o más estructuras sintácticas de parámetros HRD pueden ocurrir en orden de codificación en el VPS antes que la estructura sintáctica de parámetros HRD particular. En respuesta a determinar que la estructura sintáctica de parámetros HRD particular incluye el conjunto de parámetros HRD que son comunes a cada sub-capa del flujo de bits, el dispositivo puede realizar una operación utilizando la estructura sintáctica de parámetros HRD particular, incluido el conjunto de parámetros HRD que son comunes a cada sub-capa del flujo de bits.

[0033] Además, un codificador de vídeo puede generar mensajes SEI de anidado escalables. Un mensaje SEI de anidado escalable contiene uno o más mensajes SEI. Los mensajes SEI de anidado en un mensaje SEI de anidado escalable pueden incluir parámetros HRD u otra información asociada con un punto de funcionamiento. Algunas propuestas para HEVC no permiten que se aplique un mensaje SEI a múltiples puntos de funcionamiento. Esto puede disminuir la eficiencia de la velocidad de transferencia de bits porque puede hacer que un codificador de vídeo indique múltiples mensajes SEI con la misma información. En consecuencia, las técnicas de esta divulgación pueden permitir que un mensaje SEI se aplique a múltiples puntos de funcionamiento. Por ejemplo, un mensaje SEI de anidado escalable puede incluir elementos sintácticos que especifiquen múltiples puntos de funcionamiento aplicables a mensajes SEI anidados dentro del mensaje SEI de anidado escalable.

[0034] Además, las unidades SEI NAL, como otros tipos de unidades NAL, incluyen una cabecera de la unidad NAL y un cuerpo de la unidad NAL. El cuerpo de la unidad NAL de una unidad SEI NAL puede incluir un mensaje SEI, tal como un mensaje SEI de anidado escalable u otro tipo de mensaje SEI. Al igual que otras unidades NAL, la cabecera de la unidad NAL de una unidad SEI NAL puede incluir un elemento sintáctico `nuh_reserved_zero_6bits` y un elemento sintáctico `nuh_temporal_id_plus1`. Sin embargo, en algunas propuestas para HEVC, el elemento sintáctico `nuh_reserved_zero_6bits` y/o el elemento sintáctico `nuh_temporal_id_plus1` de una cabecera de la unidad NAL de una unidad SEI NAL no se utilizan para determinar un punto de funcionamiento aplicable a un mensaje SEI (o mensajes SEI) encapsulado por la unidad SEI NAL. Sin embargo, estos elementos sintácticos de la cabecera de la unidad SEI NAL podrían reutilizarse para reducir el número de bits indicados. Por lo tanto, de acuerdo con las técnicas de esta divulgación, un elemento sintáctico puede indicarse en un mensaje SEI de anidado escalable para indicar si un punto de funcionamiento aplicable a mensajes SEI de anidado en una unidad SEI NAL es el punto de funcionamiento indicado por la información de identificación de capa en la cabecera de la unidad NAL de la unidad SEI NAL. La información de identificación de capa en la cabecera de la unidad NAP de una unidad SEI NAL puede incluir el valor de `nuh_reserved_zero_6bits` y el valor de `nuh_temporal_id_plus1` de la cabecera de la unidad NAL.

[0035] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo de ejemplo 10 que puede utilizar las técnicas de esta divulgación. Tal como se utiliza en el presente documento, el término "codificador de vídeo" se refiere genéricamente tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" pueden referirse genéricamente a la codificación de vídeo o la descodificación de vídeo.

[0036] Como se muestra en la FIG. 1, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 y un dispositivo de destino 14. El dispositivo de origen 12 genera datos de vídeo codificados. En consecuencia, el dispositivo de origen 12 puede denominarse dispositivo de codificación de vídeo o aparato de codificación de vídeo. El dispositivo de destino 14 puede descodificar los datos de vídeo codificados generados por el dispositivo de origen 12. En consecuencia, el dispositivo de destino 14 puede denominarse dispositivo de descodificación de vídeo o aparato de descodificación de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 pueden ser ejemplos de dispositivos de codificación de vídeo o aparatos de codificación de vídeo.

[0037] El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia variedad de dispositivos, incluyendo ordenadores de sobremesa, dispositivos informáticos móviles, notebooks (es decir, portátiles), ordenadores tipo tablet, descodificadores, equipos telefónicos portátiles tales como los denominados teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores de coche o similares.

[0038] El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 a través de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicaciones que permiten al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de origen 12 puede modular los datos de vídeo codificados de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. El uno o más medios de comunicaciones pueden incluir medios de comunicaciones inalámbricos y/o cableados, tales como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El uno o más medios de comunicación pueden formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global (por ejemplo, Internet). El uno o más medios de comunicación pueden incluir routers, conmutadores, estaciones base u otros equipos que faciliten la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

[0039] En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacena los datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento, por ejemplo a través del acceso al disco o el acceso a la tarjeta. El medio de almacenamiento puede incluir una variedad de medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria flash u otros medios adecuados de almacenamiento digital para almacenar datos de vídeo codificados.

[0040] En un ejemplo adicional, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacene los datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder a datos de vídeo codificados almacenados en el servidor de archivos u otro dispositivo de almacenamiento intermedio a través de transmisión o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir los datos de vídeo codificados al dispositivo de destino 14. Entre los ejemplos de servidores de archivos se incluyen servidores web (por ejemplo, para un sitio web), servidores de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados a la red (NAS) y unidades de disco local.

[0041] El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de una conexión de datos estándar, como una conexión a Internet. Entre los ejemplos de tipos de conexiones de datos pueden incluirse canales inalámbricos (por ejemplo, conexiones Wi-Fi), conexiones alámbricas (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión sin descarga, una transmisión de descarga o una combinación de ambas.

[0042] Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de una diversidad de aplicaciones multimedia, tales como radiodifusiones de televisión por el aire, radiodifusiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, mediante Internet, codificación de datos de vídeo para su almacenamiento en un medio de almacenamiento de datos, descodificación de datos de vídeo almacenados en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 puede configurarse para soportar transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tales como radiodifusión de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

[0043] La FIG. 1 es meramente un ejemplo y las técnicas de esta divulgación pueden aplicarse a ajustes de codificación de vídeo (por ejemplo, codificación de vídeo o descodificación de vídeo) que no incluyen necesariamente ninguna comunicación de datos entre los dispositivos de codificación y descodificación. En otros ejemplos, los datos se recuperan de una memoria local, se transmiten a través de una red o similar. Un dispositivo de codificación de vídeo puede codificar y almacenar datos en la memoria, y/o un dispositivo de descodificación de vídeo puede recuperar y descodificar datos de la memoria. En muchos ejemplos, la codificación y descodificación se realiza mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en la memoria y/o recuperan y descodifican datos de la memoria.

[0044] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos ejemplos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. La fuente de vídeo 18 puede incluir un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene datos de vídeo previamente capturados, una interfaz de alimentación de vídeo para recibir datos de vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de vídeo, o una combinación de dichas fuentes de datos de vídeo.

[0045] El codificador de vídeo 20 puede codificar datos de vídeo de la fuente de vídeo 18. En algunos ejemplos, el dispositivo de origen 12 directamente transmite los datos de vídeo codificados al dispositivo destino 14 a través de un canal de comunicación 22. En otros ejemplos, los datos de vídeo codificados también pueden almacenarse en un medio de almacenamiento o un servidor de archivos para un acceso posterior mediante el dispositivo de destino 14 para la descodificación y/o reproducción.

[0046] En el ejemplo de la FIG. 1, el dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos ejemplos, la interfaz de entrada 28 incluye un receptor y/o un módem. La interfaz de entrada 28 puede recibir los datos de vídeo codificados por el canal 16. El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En general, el dispositivo de visualización 32 muestra datos de vídeo descodificados. El dispositivo de visualización 32 puede comprender una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0047] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera entre una variedad de circuitos adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), conjuntos de puertas programables sobre el terreno

(FPGA), lógica discreta, hardware o cualquier combinación de los mismos. Si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio de almacenamiento legible por ordenador no transitorio adecuado, y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cualquiera de los anteriores (incluyendo hardware, software, una combinación de hardware y software, etc.) puede considerarse como uno o más procesadores. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

[0048] Esta divulgación puede referirse en general al codificador de vídeo 20 que "indica" cierta información a otro dispositivo, tal como el descodificador de vídeo 30. El término "indicar" puede referirse en general a la comunicación de elementos sintácticos y/u otros datos usados para descodificar los datos de vídeo comprimidos. Dicha comunicación puede producirse en tiempo real o casi real. De forma alternativa, dicha comunicación puede producirse durante un lapso de tiempo, tal como podría ocurrir cuando se almacenan elementos sintácticos en un medio de almacenamiento legible por ordenador en un flujo de bits codificado en el momento de la codificación, que a continuación un dispositivo de descodificación puede recuperar en cualquier momento tras haber sido almacenado en este medio.

[0049] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 funcionan de acuerdo con una norma de compresión de vídeo, tal como ISO/IEC MPEG-4 Visual e ITU-T H.264. (también conocida como ISO/IEC MPEG-4 AVC), incluida su ampliación de codificación de vídeo escalable (SVC), ampliación de codificación de vídeo múltiples visualizaciones (MVC) y ampliación 3DV basada en MVC. En algunos casos, cualquier flujo de bits que se ajuste a 3DV basada en MVC siempre contiene un sub-flujo de bits que es compatible con un perfil MVC, por ejemplo, alto perfil estéreo. Además, actualmente se están emprendiendo unas iniciativas para generar una ampliación de codificación de vídeo tridimensional (3DV) para H.264/AVC, en concreto, 3DV basada en AVC. En otros ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con la Recomendación ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC PEG-2 Visual e ITU-T H.264, ISO/IEC Visual.

[0050] En otros ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con la norma de Codificación de Vídeo de Alta Eficiencia (HEVC) elaborada actualmente por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC. Un borrador de la próxima norma HEVC, denominado "Borrador de Trabajo 8 de HEVC", se describe en el documento de Bross et al., "Codificación de Vídeo de Alta Eficiencia (HEVC), borrador de memoria descriptiva textual 8", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 10.ª Reunión, Estocolmo, Suecia, julio de 2012, que a partir del 13 de junio de 2013, está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip. Otro borrador de la próxima norma HEVC, denominado "Borrador de Trabajo 9 de la HEVC" se describe en el documento de Bross et al. "Borrador 9 de Memoria Descriptiva Textual de Codificación de Vídeo de Alta Eficiencia (HEVC)", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 11.ª reunión, Shanghái, China, octubre de 2012, que a partir del 13 de junio de 2013 está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v13.zip. Además, actualmente se están dedicando esfuerzos a la elaboración de ampliaciones SVC, MVC y 3DV para HEVC. La ampliación 3DV de HEVC puede denominarse 3DV basada en HEVC, o HEVC-3DV.

[0051] En HEVC y otras normas de codificación de vídeo, una secuencia de vídeo incluye típicamente una serie de imágenes. Las imágenes también pueden denominarse "tramas". Una imagen puede incluir tres matrices de muestras, denominadas S_L , S_{Cb} y S_{Cr} . S_L es una matriz bidimensional (es decir, un bloque) de muestras de luma. S_{Cb} es una matriz bidimensional de muestras de crominancia Cb. S_{Cr} es una matriz bidimensional en muestras de crominancia Cr. Las muestras de crominancia también se pueden denominar en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede solo incluir una matriz de muestras de luma.

[0052] Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede ser un bloque de árbol de codificación de muestras de luma, dos bloques de árbol de codificación correspondientes de muestras de croma y estructuras sintácticas usadas para codificar las muestras de los bloques de codificación. Un bloque de árbol de codificación puede ser un bloque NxN de muestras. Una CTU también puede denominarse "bloque de árbol" o una "unidad de codificación más grande" (LCU). Las CTU de HEVC pueden ser análogas en gran medida a los macrobloques de otras normas, tales como la H.264/AVC. Sin embargo, una CTU no está limitada necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un fragmento puede incluir un número entero de CTU ordenadas consecutivamente por orden de codificación.

[0053] Para generar una CTU codificada, el codificador de vídeo 20 puede llevar a cabo de forma recursiva una partición de árbol cuaternario en los bloques de árbol de codificación de una CTU para dividir los bloques de árbol de codificación en bloques de codificación, de ahí el nombre "unidades de árbol de codificación." Un bloque de

codificación es un bloque NxN de muestras. Una CU puede ser un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene una matriz de muestras de luma, una matriz de muestras de Cb y una matriz de muestras de Cr y estructuras sintácticas usadas para codificar las muestras de los bloques de codificación. El codificador de vídeo 20 puede dividir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción puede ser un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las que se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede ser un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma de una imagen y estructuras sintácticas usadas para predecir las muestras de bloques de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU.

[0054] El codificador de vídeo 20 puede usar intra-predicción o inter-predicción para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 utiliza intra-predicción para generar el bloque de píxeles predictivos de una PU, el codificador de vídeo 20 puede generar los bloques de píxeles predictivos de la PU basándose en muestras descodificadas de la imagen asociada con la PU.

[0055] Si el codificador de vídeo 20 utiliza inter-predicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU basándose en muestras descodificadas de una o más imágenes distintas a la imagen asociada con la PU. El codificador de vídeo 20 puede usar uni-predicción o bi-predicción para generar los bloques predictivos de una PU. Cuando el codificador de vídeo 20 usa la predicción unidimensional para generar los bloques predictivos para una PU, la PU puede tener un solo vector de movimiento. Cuando el codificador de vídeo 20 usa predicción bidireccional para generar los bloques predictivos para una PU, la PU puede tener dos vectores de movimiento.

[0056] Después de que el codificador de vídeo 20 genera luma predictivo, bloques Cb y Cr para una o más PU de una CU, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra en el bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques predictivos de luma de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual de Cb para la CU. Cada muestra en el bloque residual Cb de CU puede indicar una diferencia entre una muestra Cb en uno de los bloques Cb predictivos de CU y una muestra correspondiente en el bloque de codificación Cb original de CU. El codificador de vídeo 20 también puede generar un bloque residual de Cr para la CU. Cada muestra en el bloque residual Cr de CU puede indicar una diferencia entre una muestra Cr en uno de los bloques Cr predictivos de CU y una muestra correspondiente en el bloque de codificación Cr original de CU.

[0057] Por otra parte, el codificador de vídeo 20 puede usar la partición en árbol cuádruple para descomponer los bloques residuales luma, Cb y Cr de una CU en uno o más bloques de transformada de luma, Cb y Cr. Un bloque de transformada puede ser un bloque rectangular de muestras en el que se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede ser un bloque de transformada de muestras de luma, dos bloques de transformada correspondientes de muestras de croma y estructuras sintácticas utilizadas para transformar las muestras de bloques de transformada. De este modo, cada TU de una CU puede estar asociada con un bloque de transformada de luma, un bloque de transformada de Cb y un bloque de transformada de Cr. El bloque de transformada de luma asociado con la TU puede ser un sub-bloque del bloque residual de luma de la CU. El bloque de transformada de Cb puede ser un sub-bloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un sub-bloque del bloque residual de Cr de la CU.

[0058] El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU con el fin de generar un bloque de coeficientes luma para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cb de una TU para generar un bloque de coeficientes para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

[0059] Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes luma, un bloque de coeficientes Cb o un bloque de coeficientes Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación se refiere en general a un proceso en el que los coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes de transformada, proporcionando una compresión adicional. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos sintácticos que indican los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar Codificación Aritmética Binaria Adaptable al Contexto (CABAC) en los elementos sintácticos que indican los coeficientes de transformada cuantificados. El codificador de vídeo 20 puede emitir los elementos sintácticos codificados por entropía, en un flujo de bits.

[0060] El codificador de vídeo 20 puede emitir un flujo de bits que incluye los elementos sintácticos codificados por

entropía. El flujo de bits puede incluir una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL incluye una cabecera de la unidad NAL y encapsula una carga útil de secuencia de bytes sin procesar (RBSP). La cabecera de la unidad NAL puede incluir un elemento sintáctico que indica un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por la cabecera de la unidad NAL de una unidad NAL indica el tipo de la unidad NAL. Un RBSP puede ser una estructura sintáctica que contiene un número entero de bytes que está encapsulado dentro de una unidad NAL. En algunos casos, una RBSP incluye cero bits.

[0061] Diferentes tipos de unidades NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad NAL puede encapsular una RBSP para un conjunto de parámetros de imagen (PPS), un segundo tipo de unidad NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad NAL puede encapsular una RBSP para SEI, etc. Las unidades NAL que encapsulan las RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes SEI) pueden denominarse unidades NAL de capa de codificación de vídeo (VCL).

[0062] El descodificador de vídeo 30 puede recibir el flujo de bits generado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede analizar el flujo de bits para descodificar elementos sintácticos del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose al menos en parte en los elementos sintácticos descodificados del flujo de bits. El proceso para reconstruir los datos de vídeo puede ser en general recíproco al proceso realizado por el codificador de vídeo 20. Por ejemplo, el descodificador de vídeo 30 puede usar vectores de movimiento de PU para determinar bloques predictivos para las PU de una CU actual. Además, el descodificador de vídeo 30 puede cuantificar inversamente los bloques de coeficientes de transformada asociados con las TU de la CU actual. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes de transformada para reconstruir bloques de transformada asociados con las TU del CM actual. El descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques de predicción para las PU de la CU actual a muestras correspondientes de los bloques de transformada de las TU de la CU actual. Mediante la reconstrucción de los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

[0063] En la codificación de múltiples visualizaciones, puede haber varias visualizaciones de la misma escena desde diferentes puntos de visualización. El término "unidad de acceso" se utiliza para referirse al conjunto de imágenes que corresponden a la misma instancia de tiempo. Así, los datos de vídeo pueden conceptualizarse como una serie de unidades de acceso que se producen a lo largo del tiempo. Un "componente de visualización" puede ser una representación codificada de una visualización en una única unidad de acceso. En esta divulgación, una "visualización" puede referirse a una secuencia de componentes de visualización asociadas con el mismo identificador de visualización.

[0064] La codificación de múltiples visualizaciones soporta predicción entre visualizaciones. La predicción entre visualizaciones es similar a la inter-predicción utilizada en HEVC y puede usar los mismos elementos sintácticos. Sin embargo, cuando un codificador de vídeo realiza predicción entre visualizaciones en una unidad de vídeo actual (como una PU), el codificador de vídeo 20 puede usar, como una imagen de referencia, una imagen que está en la misma unidad de acceso que la unidad de vídeo actual, pero en una visualización diferente. En contraste, la inter-predicción convencional solo usa imágenes en diferentes unidades de acceso como imágenes de referencia.

[0065] En la codificación de múltiples visualizaciones, una visualización puede denominarse "visualización básica" si un descodificador de vídeo (por ejemplo, descodificador de vídeo 30) puede descodificar imágenes de la visualización sin referencia a imágenes en ninguna otra visualización. Cuando se codifica una imagen en una de las visualizaciones no básicas, un codificador de vídeo (como el codificador de vídeo 20 o el descodificador de vídeo 30) puede agregar una imagen a una lista de imágenes de referencia si la imagen está en una visualización diferente pero en la misma instancia de tiempo (es decir, unidad de acceso) que la imagen que el codificador de vídeo está codificando actualmente. Al igual que otras imágenes de referencia de inter-predicción, el codificador de vídeo puede insertar una imagen de referencia de predicción entre visualizaciones en cualquier posición de una lista de imágenes de referencia.

[0066] Las normas de codificación de vídeo especifican modelos de almacenamiento en memoria intermedia de vídeo. En H.264/AVC y HEVC, el modelo de almacenamiento en memoria intermedia se denomina "descodificador de referencia hipotético" o "HRD". En el Borrador de Trabajo 8 de HEVC, el HRD se describe en el Anexo C.

[0067] El HRD describe cómo los datos se almacenan en memoria intermedia para la descodificación y cómo los datos descodificados se almacenan en memoria intermedia para la salida. Por ejemplo, el HRD describe el funcionamiento de un CPB, una memoria intermedia de imágenes descodificadas ("DPB") y un proceso de descodificación de vídeo. El CPB es una memoria intermedia de primero en entrar, primero en salir, que contiene unidades de acceso en un orden de descodificación especificado por HRD. El DPB es una memoria intermedia que contiene imágenes descodificadas para referencia, reordenamiento de salida o retardo de salida especificado por el HRD. Los comportamientos del CPB y DPB se pueden especificar matemáticamente. El HRD puede imponer

directamente restricciones en el tiempo, los tamaños de la memoria intermedia y las velocidades de transferencia de bits. Además, el HRD puede imponer restricciones indirectamente a varias características y estadísticas de flujo de bits.

5 **[0068]** En H.264/AVC y HEVC, la conformidad del flujo de bits y la conformidad del descodificador se especifican como partes de la especificación del HRD. En otras palabras, el modelo de HRD especifica pruebas para determinar si un flujo de bits cumple con la norma y pruebas para determinar si un descodificador cumple con la norma. Aunque el HRD se denomina como algún tipo de descodificador, los codificadores de vídeo típicamente usan el HRD para garantizar la conformidad del flujo de bits, mientras que los descodificadores de vídeo típicamente no necesitan el HRD.

10 **[0069]** H.264/AVC y HEVC especifican dos tipos de conformidad de flujo de bits o de HRD; esto es, Tipo I y Tipo II. Un flujo de bits de Tipo I es un flujo de unidad NAL que contiene solo las unidades NAL VCL y la unidad NAL de datos de relleno para todas las unidades de acceso en el flujo de bits. Un flujo de bits de Tipo II es un flujo de unidad NAL que contiene, además de las unidades NAL VCL y unidades NAL de datos de relleno para todas las unidades de acceso en el flujo de bits, al menos uno de los siguientes: unidades NAL no VCL adicionales distintas de unidades NAL de datos de relleno; y todos los elementos sintácticos `leading_zero_8bits`, `zero_byte`, `start_coded_prefix_one_3bytes` y `trailing_zero_8bits` que forman una secuencia de bytes desde la secuencia de la unidad NAL.

15 **[0070]** Cuando un dispositivo realiza una conformidad de flujo de bits, prueba que determina si un flujo de bits se ajusta a una norma de codificación de vídeo, el dispositivo puede seleccionar un punto de funcionamiento del flujo de bits. El dispositivo puede entonces determinar un conjunto de parámetros HRD aplicables al punto de funcionamiento seleccionado. El dispositivo puede usar el conjunto de parámetros HRD aplicable al punto de funcionamiento seleccionado para configurar el comportamiento del HRD. Más particularmente, el dispositivo puede usar el conjunto aplicable de parámetros HRD para configurar los comportamientos de componentes particulares del HRD, tales como un planificador de flujo hipotético (HSS), el CPB, un proceso de descodificación, el DPB, etc. Posteriormente, el HSS puede inyectar datos de vídeo codificados del flujo de bits en el CPB del HRD de acuerdo con un programa particular. Además, el dispositivo puede invocar un proceso de descodificación que descodifica los datos de vídeo codificados en el CPB. El proceso de descodificación puede enviar imágenes descodificadas al DPB. A medida que el dispositivo mueve datos a través del HRD, el dispositivo puede determinar si un conjunto particular de restricciones sigue siendo satisfactorio. Por ejemplo, el dispositivo puede determinar si se produce una condición de sobreflujo o sub-flujo en el CPB o DPB mientras el HRD está descodificando la representación del punto de funcionamiento del punto de funcionamiento seleccionado. El dispositivo puede seleccionar y procesar cada punto de funcionamiento del flujo de bits de esta manera. Si ningún punto de funcionamiento del flujo de bits hace que se violen las restricciones, el dispositivo puede determinar que el flujo de bits se ajusta a la norma de codificación de vídeo.

20 **[0071]** H.264/AVC y HEVC especifican dos tipos de conformidad de descodificador; esto es, la conformidad de descodificador de sincronización de salida y la conformidad de descodificador del orden de salida. Un descodificador que alega conformidad, con un perfil, fase y nivel específico, es capaz de descodificar con éxito todos los flujos de bits que se ajustan a los requisitos de conformidad de flujo de bits de una norma de codificación de vídeo, como HEVC. En esta divulgación, un "perfil" puede referirse a un subconjunto de la sintaxis del flujo de bits. "Fases" y "niveles" se pueden especificar dentro de cada perfil. Un nivel de una fase puede ser un conjunto de restricciones especificadas impuestas en los valores de los elementos sintácticos en el flujo de bits. Estas limitaciones pueden ser simples limitaciones de valores. De forma alternativa, las restricciones pueden tomar la forma de restricciones sobre combinaciones aritméticas de valores (por ejemplo, el ancho de imagen multiplicado por la altura de imagen multiplicada por el número de imágenes descodificadas por segundo). Típicamente, un nivel especificado para una fase inferior es más limitado que un nivel especificado para una fase superior.

25 **[0072]** Cuando un dispositivo realiza una prueba de conformidad de descodificador para determinar si un descodificador bajo prueba (DUT) se ajusta a una norma de codificación de vídeo, el dispositivo puede proporcionar, tanto al HRD y el DUT, un flujo de bits que se ajusta a la norma de codificación de vídeo. El HRD puede procesar el flujo de bits de la manera descrita anteriormente con respecto a la prueba de conformidad de flujo de bits. El dispositivo puede determinar que el DUT se ajusta a la norma de codificación de vídeo si el orden de las imágenes descodificadas emitidas por el DUT coincide con el orden de las imágenes descodificadas emitidas por el HRD. Además, el dispositivo puede determinar que el DUT se ajusta a la norma de codificación de vídeo si el tiempo con el cual el DUT envía imágenes descodificadas coincide con el tiempo con el cual el HRD emite las imágenes descodificadas.

30 **[0073]** En los modelos H.264/AVC y HEVC HRD, la descodificación o eliminación de CPB pueden basarse en unidad de acceso. Es decir, se supone que el HRD descodifica unidades de acceso completas de una vez y elimina unidades de acceso completas del CPB. Además, en los modelos H.264/AVC y HEVC HRD, se supone que la descodificación de imágenes es instantánea. El codificador de vídeo 20 puede indicar, en los mensajes SEI de sincronización de imagen, los tiempos de descodificación para comenzar la descodificación de las unidades de acceso. En aplicaciones prácticas, si un descodificador de vídeo en conformidad sigue estrictamente los tiempos de

descodificación indicados para iniciar la descodificación de unidades de acceso, el primer momento posible para emitir una imagen descodificada específica es igual al momento de descodificación de esa imagen específica, más el tiempo necesario para descodificar esa imagen específica. Sin embargo, en el mundo real, el tiempo necesario para descodificar una imagen no puede ser igual a cero.

5 **[0074]** Los parámetros HRD pueden controlar diversos aspectos del HRD. En otras palabras, el HRD puede confiar en los parámetros HRD. Entre los parámetros HRD pueden incluirse un retardo inicial de eliminación de CPB, un tamaño de CPB, una velocidad de transferencia de bits, un retardo inicial de salida de DPB y un tamaño de DPB. El codificador de vídeo 20 puede indicar estos parámetros HRD en una estructura sintáctica `hrd_parameters()` especificada en un conjunto de parámetros de vídeo (VPS) y/o un conjunto de parámetros de secuencia (SPS). Los VPS y/o SPS individuales pueden incluir múltiples estructuras sintácticas `hrd_parameters()` para diferentes conjuntos de parámetros HRD. En algunos ejemplos, el codificador de vídeo 20 puede indicar parámetros HRD en mensajes SEI de período de almacenamiento en memoria intermedia o mensajes SEI de sincronización de imágenes.

15 **[0075]** Como se explicó anteriormente, un punto de funcionamiento de un flujo de bits está asociado con un conjunto de identificadores de capa (es decir, un conjunto de valores `nuh_reserved_zero_6bits`) y un identificador temporal. Una representación del punto de funcionamiento puede incluir cada unidad NAL que está asociada con el punto de funcionamiento. Una representación de punto de funcionamiento puede tener una velocidad de tramas y/o velocidad de transferencia de bits diferente a la de un flujo de bits original. Esto se debe a que la representación del punto de funcionamiento puede no incluir algunas imágenes y/o algunos de los datos del flujo de bits original. Por lo tanto, si el descodificador de vídeo 30 fuera a eliminar datos del CPB, y/o el CPB a una velocidad particular cuando procesa el flujo de bits original y si el descodificador de vídeo 30 fuera a eliminar datos del CPB y/o el DPB a la misma velocidad cuando procesa una representación de punto de funcionamiento, el descodificador de vídeo 30 puede eliminar demasiados o muy pocos datos del CPB y/o el DPB. En consecuencia, el codificador de vídeo 20 puede indicar diferentes conjuntos de parámetros HRD para diferentes puntos de funcionamiento. Por ejemplo, el codificador de vídeo 20 puede incluir, en un VPS, múltiples estructuras sintácticas `hrd_parameters()` que incluyen parámetros HRD para diferentes puntos de funcionamiento.

30 **[0076]** En el Borrador de Trabajo 8 de HEVC, el conjunto de parámetros HRD incluye opcionalmente un conjunto de información que es común para todas las sub-capas temporales. En otras palabras, el conjunto de parámetros HRD puede incluir opcionalmente un conjunto de elementos sintácticos comunes que son aplicables a puntos de funcionamiento que incluyen cualquier sub-capa temporal. Una sub-capa temporal puede ser una capa temporal escalable de un flujo de bits temporal escalable que consta de unidades NAL VCL con un valor particular de `TemporalId` y las unidades NAL no VCL asociadas. Además del conjunto de información común, los conjuntos de parámetros HRD pueden incluir un conjunto de elementos sintácticos que son específicos de sub-capas temporales individuales. Por ejemplo, la estructura sintáctica `hrd_parameters()` puede incluir opcionalmente un conjunto de información que es común para todas las sub-capas y siempre incluye información específica de sub-capas. Debido a que el conjunto de información común es común a múltiples conjuntos de parámetros HRD, puede ser innecesario indicar el conjunto de información común en múltiples conjuntos de parámetros HRD. Por el contrario, en el Borrador de Trabajo 8 de HEVC, la información común puede estar presente en un conjunto de parámetros HRD cuando el conjunto de parámetros HRD es el primer conjunto de parámetros HRD en un VPS o la información común puede estar presente en un conjunto de parámetros HRD cuando el conjunto de parámetros HRD está asociado con un primer índice de punto de funcionamiento. Por ejemplo, HEVC. El Borrador de Trabajo 8 soporta la presencia de información común cuando la estructura sintáctica `hrd_parameters()` es la primera estructura sintáctica `hrd_parameters()` en el VPS o cuando la estructura sintáctica `hrd_parameters()` está asociada con un primer índice de punto de funcionamiento.

45 **[0077]** La Tabla 1, a continuación, es una estructura sintáctica de ejemplo para una estructura sintáctica `hrd_parameters()` en HEVC.

50

TABLA 1- Parámetros HRD

hrd_parameters(commonInfPresentFlag, MaxNumSubLayersMinus1) {	Descriptor
if(commonInfPresentFlag) {	
timing_info_present_flag	u(1)
if(timing_info_present_flag) {	
num_units_it_tick	u(32)
time_scale	u(32)
}	
nal_hrd_parameters_present_flag	u(1)
vcl_hrd_parameters_Present_flag	u(1)
if(nal_hrd_parameters_present_flag vcl_hrd_parameters_Present_flag){	
sub_pic_cpb_params_present_flag	u(1)
if(sub_pic_cpb_params_present_flag) {	
tick_divisor_minus2	u(8)

hrd_parameters(commonInfPresentFlag, MaxNumSubLayersMinus1) {	Descriptor
du_cpb_removal_delay_length_minus1	u(5)
}	
bit_rate_scale	u(4)
cpb_size_scale	u(4)
initial_cpb_removal_delay_length_minus1	u(5)
cpb_removal_delay_length_minus1	u(5)
dpb_output_delay_length_minus1	u(5)
}	
}	
for(i=0; i <= MaxNumSubLayersMinus1; i++) {	
fixed_pic_rate_flag[i]	u(1)
fixed_pic_rate_flag[i]	
pic_duration_in_tc_minus1[i]	ue(v)
low_delay_hrd_flag[i]	u(1)
cpb_cnt_minus1[i]	ue(v)
if(nal_hrd_parameters_present_flag)	
sub_layer_hrd_parameters(i)	
if(vcl_hrd_parameters_present_flag)	
sub_layer_hrd_parameters(i)	
}	
}	

5 **[0078]** En el ejemplo de la Tabla 1 anterior y en otras tablas sintácticas de esta divulgación, los elementos sintácticos con descriptor de tipo ue(v) pueden ser números enteros de longitud variable sin signo codificados utilizando codificación Golomb exponencial (Exp-golomb) de orden 0 con el bit izquierdo primero. En el ejemplo de la Tabla 1 y las siguientes tablas, los elementos sintácticos que tienen descriptores de la forma u(n), donde n es un entero no negativo, son valores sin signo de longitud n.

10 **[0079]** En el ejemplo de sintaxis de la Tabla 1, los elementos sintácticos en el bloque “if (commonInfPresentFlag {...}” son la información común de las estructuras sintácticas de parámetros HRD. En otras palabras, la información común del conjunto de parámetros HRD puede incluir los elementos sintácticos timing_info_present_flag, num_units_in_tick, time_scale, nal_hrd_parameters_present_flag, vcl_hrd_parameters_present_flag, sub_pic_cpb_params_present_flag, tick_divisor_minus2, du_cpb_removal_delay_length_minus1, bit_rate_scale, cpb_size_scale, initial_cpb_removal_delay_length_minus1, cpb_removal_delay_length_minus1 y dpb_output_delay_length_minus1.

15 **[0080]** Además, en el ejemplo de la Tabla 1, los elementos sintácticos fixed_pic_rate_flag[i], pic_duration_in_tc_minus1[i], low_delay_hrd_flag[i] y cpb_cnt_minus1[i] pueden ser un conjunto de parámetros HRD específicos de sub-capa. En otras palabras, estos elementos sintácticos de la estructura sintáctica hrd_parameter() solo pueden aplicarse a los puntos de funcionamiento que incluyen una sub-capa específica. Por lo tanto, los parámetros HRD de una estructura sintáctica hrd_parameters() pueden incluir, además de la información común incluida opcionalmente, un conjunto de parámetros HRD específicos de sub-capa que son específicos de una sub-capa particular del flujo de bits.

25 **[0081]** El elemento sintáctico fixed_pic_rate_flag[i] puede indicar que, cuando HighestTid es igual a i, la distancia temporal entre los tiempos de salida de HRD de cualesquiera dos imágenes consecutivas en orden de salida está limitado de manera específica. HighestTid puede ser una variable que identifica una sub-capa temporal más alta (por ejemplo, de un punto de funcionamiento). El elemento sintáctico pic_duration_in_tc_minus1[i] puede especificar, cuando HighestTid es igual a i, la distancia temporal, en pulsos de reloj, entre los tiempos de salida de HRD de cualquier imagen consecutiva en orden de salida en la secuencia de vídeo codificada. El elemento sintáctico low_delay_hrd_flag[i] puede especificar el modo de funcionamiento de HRD, cuando HighestTid es igual a i, como se especifica en el Anexo C del Borrador de Trabajo 8 de HEVC. El elemento sintáctico cpb_cnt_minus1[i] puede especificar el número de especificaciones de CPB alternativas en el flujo de bits de la secuencia de vídeo codificada cuando HighestTid es igual a i, en el que una especificación de CPB alternativa se refiere a una operación de CPB particular con un conjunto particular de parámetros de CPB.

35 **[0082]** El codificador de vídeo 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 las imágenes. Sin embargo, el decodificador de vídeo 30 u otros dispositivos pueden usar los metadatos incluidos en los mensajes SEI para otros fines diversos. Por ejemplo, el decodificador de vídeo 30 u otro dispositivo puede usar los metadatos en los mensajes SEI para la sincronización de salida de la imagen, visualización de la imagen, detección de pérdida y ocultación de errores.

40 **[0083]** El codificador de vídeo 20 puede incluir una o más unidades SEI NAL en una unidad de acceso. En otras

palabras, cualquier número de unidades SEI NAL puede estar asociada con una unidad de acceso. Además, cada unidad SEI NAL puede contener uno o más mensajes SEI. La norma HEVC describe la sintaxis y la semántica para varios tipos de mensajes SEI. Sin embargo, la norma HEVC no describe el manejo de los mensajes SEI porque los mensajes SEI no afectan el proceso de descodificación normativa. Una razón para tener mensajes SEI en la norma HEVC es permitir que los datos suplementarios se interpreten de manera idéntica en diferentes sistemas usando HEVC. Las especificaciones y sistemas que usan HEVC pueden requerir codificadores de vídeo para generar ciertos mensajes SEI o pueden definir el manejo específico de tipos particulares de mensajes SEI recibidos. La Tabla 2, a continuación, enumera los mensajes SEI especificados en HEVC y describe brevemente sus propósitos.

TABLA 2 - Visión general de mensajes SEI

Mensaje SEI	Finalidad
Período de almacenamiento temporal	Retardos iniciales para la operación del descodificador de referencia hipotética (HRD)
Sincronización de imágenes	Tiempo de salida de la imagen y tiempo de eliminación de imagen / sub-imagen para el funcionamiento del HRD
Rectángulo de escaneo panorámico	Exhibición con una relación de aspecto de imagen (PAR) diferente a la PAR de las imágenes de salida
Carga útil de relleno	Ajuste de la velocidad de transferencia de bits para satisfacer limitaciones específicas
Datos de usuario registrados - Datos de usuario no registrados	Mensajes SEI a especificar por entidades externas
Punto de recuperación	Información adicional para acceso aleatorio puro. Actualización gradual de descodificación.
información de escena	Información sobre cambios de escena y transiciones
Instantánea a pantalla completa	Indicación para etiquetar la imagen descodificada asociada como una instantánea de imagen fija del contenido de vídeo
Segmento de refinamiento progresivo	Indica que ciertas imágenes consecutivas representan un refinamiento progresivo de la calidad de una imagen, en lugar de una escena en movimiento
Características de textura de película	Permite a los descodificadores sintetizar la textura de la película
Preferencia de exhibición de filtro desbloqueador	Recomienda si las imágenes exhibidas deberían someterse o no al proceso del filtro desbloqueador de bucle interno
Sugerencia de pos-filtro	Proporciona coeficientes de pos-filtro sugeridos o información de correlación para el diseño de pos-filtro
Información de correlación de tonos	Re-correlación con otro espacio de colores distinto al usado o adoptado en la codificación
Disposición de empaquetamiento de tramas	Disposición de empaquetamiento de tramas Empaquetamiento de vídeo estereoscópico en un flujo de bits de HEVC
Orientación de la pantalla	Especifica el vuelco y/o la rotación que deberían aplicarse a las imágenes de salida cuando se exhiben
Indicación de campo	Proporciona información relacionada con el contenido de vídeo intercalado y/o la codificación de campo; por ejemplo, indica si la imagen es una trama progresiva, un campo o una trama que contiene dos campos intercalados
Información no válida de imagen descodificada	Suma de comprobación de la imagen descodificada, que puede ser utilizada para la detección de errores
Sincronización de sub-imágenes	Tiempo de eliminación de sub-imágenes para funcionamiento de HRD
Conjuntos de parámetros activos	Proporciona información sobre VPS, SPS, etc., activos
Estructura de descripción de imágenes	Describe la estructura temporal y de inter-predicción del flujo de bits

[0084] La Solicitud de Patente Provisional de Estados Unidos 61/705 102, presentada el 24 de septiembre de 2012, describe varios procedimientos para la indicación y parámetros de selección ofHRD, incluyendo indicación y selección de información de retardo y sincronización en los mensajes SEI. Hannuksela et al., "AHG9: Los puntos de funcionamiento en VPS y SEI de anidado," Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG11, 11.ª reunión, Shanghái, China, 10-19 de octubre de 2012, documento n.º JCTVC-K0180v1, que, a partir del 13 de junio de 2013, está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K0180-v1.zip, proporcionan otro procedimiento para indicación de parámetros HRD, así como un mecanismo para anidar mensajes SEI.

[0085] Hay varios problemas o deficiencias con las técnicas existentes para la indicación de parámetros HRD. Por ejemplo, las técnicas existentes pueden no permitir que un conjunto de parámetros HRD sea compartido por múltiples puntos de funcionamiento. Sin embargo, cuando el número de puntos de funcionamiento es alto, puede ser una carga para el codificador de vídeo 20 u otra unidad que intente asegurar la conformidad de un flujo de bits para

producir diferentes conjuntos de parámetros HRD para cada punto de funcionamiento. Por el contrario, la conformidad del flujo de bits puede asegurarse asegurándose de que cada punto de funcionamiento está asociado con un conjunto de parámetros HRD, pero que un conjunto particular de parámetros HRD puede ser compartido por múltiples puntos de funcionamiento. Una o más técnicas de esta divulgación pueden proporcionar un diseño para permitir que un conjunto de parámetros HRD sea compartido por múltiples puntos de funcionamiento. En otras palabras, un solo conjunto de parámetros HRD puede ser aplicable a múltiples puntos de funcionamiento. Este diseño puede permitir al codificador de vídeo 20 u otra unidad que intente garantizar la conformidad de un flujo de bits compensar la complejidad y el rendimiento.

[0086] En otro ejemplo de los problemas o deficiencias con las técnicas existentes de indicación de parámetros HRD, cuando hay múltiples conjuntos de parámetros HRD en un VPS, puede ser deseable tener múltiples conjuntos diferentes de información común para los conjuntos de parámetros HRD. Esto puede ser especialmente cierto cuando hay un gran número de estructuras sintácticas de parámetros HRD en un VPS. Por lo tanto, puede ser deseable tener conjuntos de información común en estructuras sintácticas de parámetros HRD distintas de la primera estructura sintáctica de parámetros HRD. Por ejemplo, para proporcionar un mayor rendimiento cuando hay múltiples estructuras sintácticas `hrd_parameters()` en un VPS, particularmente cuando el número total de estructuras sintácticas `hrd_parameters()` es relativamente alto, puede ser deseable tener información común diferente para estructuras sintácticas `hrd_parameters()` que no sea la información común de la primera estructura sintáctica `hrd_parameters()` o que no sea la información común del primer índice de punto de funcionamiento.

[0087] Una o más técnicas de esta divulgación proporcionan un diseño para permitir que la información común de conjuntos de parámetros HRD se indique explícitamente para cualquier conjunto de parámetros HRD. Por ejemplo, las técnicas de esta divulgación pueden permitir que la información común para todas las sub-capas sea indicada explícitamente para cualquier estructura sintáctica `hrd_parameters()`.

[0088] De esta manera, el codificador de vídeo 20 puede ser una señal, en un flujo de bits, un VPS que incluye una pluralidad de estructuras sintácticas de parámetros HRD, incluyendo cada un parámetro HRD. Para cada estructura sintáctica de parámetros HRD respectiva en la pluralidad de estructuras sintácticas de parámetros HRD, el VPS incluye además un elemento sintáctico que indica si entre los parámetros HRD de la estructura sintáctica de parámetros HRD respectiva se incluye un conjunto común de parámetros HRD además de un conjunto de información de parámetros HRD específica de sub-capa específica para una sub-capa particular del flujo de bits. El conjunto común de parámetros HRD es común a todas las sub-capas del flujo de bits.

[0089] De forma similar, el descodificador de vídeo 30 u otro dispositivo puede descodificar, a partir de un flujo de bits, un VPS que incluye una pluralidad de estructuras sintácticas de parámetros HRD, cada una de las cuales incluye parámetros HRD. Para cada estructura sintáctica de parámetros HRD respectiva en la pluralidad de estructuras sintácticas de parámetros HRD, el VPS puede incluir además un elemento sintáctico que indica si los parámetros HRD de la estructura sintáctica de parámetros HRD respectiva incluyen un conjunto común de parámetros HRD. El descodificador de vídeo 30 u otro dispositivo puede realizar una operación usando los parámetros HRD de al menos una de las estructuras sintácticas de parámetros HRD.

[0090] Además, los procedimientos existentes para el anidado de los mensajes SEI pueden tener varios problemas o deficiencias. Por ejemplo, las técnicas existentes de indicación de parámetros HRD pueden no permitir que un mensaje SEI se aplique a múltiples puntos de funcionamiento. Las técnicas de esta divulgación pueden proporcionar un diseño para permitir que un mensaje SEI se aplique a múltiples puntos de funcionamiento.

[0091] En particular, un mensaje SEI de anidado escalable puede incluir elementos sintácticos que especifican múltiples puntos de funcionamiento aplicables a los mensajes SEI de anidado en el mensaje SEI de anidado escalable. En otras palabras, los mensajes SEI de anidado escalables pueden proporcionar un mecanismo para asociar mensajes SEI con un subconjunto de flujo de bits (por ejemplo, una representación de punto de funcionamiento) o con capas y sub-capas específicas.

[0092] De esta manera, el codificador de vídeo 20 puede generar un mensaje SEI de anidado escalable que incluye una pluralidad de elementos sintácticos que identifican una pluralidad de puntos de funcionamiento a los que se aplica un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable. Además, el codificador de vídeo 20 puede indicar el mensaje SEI de anidado escalable en un flujo de bits.

[0093] De esta manera, el descodificador de vídeo 30 u otro dispositivo puede, en un proceso de codificación de vídeo, descodificar, a partir de un mensaje SEI de anidado escalable, una pluralidad de elementos sintácticos que identifican puntos de funcionamiento a los que se aplica un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable. Además, el descodificador de vídeo 30 u otro dispositivo puede realizar una operación basada al menos en parte en uno o más elementos sintácticos del mensaje SEI de anidado.

[0094] Otro ejemplo de los problemas o deficiencias con las técnicas existentes de mensajes SEI de anidado se refiere al hecho de que las técnicas existentes de mensajes SEI de anidado no utilizan el valor de un elemento sintáctico identificador de capa (por ejemplo, `nuh_reserved_zero_6bits`) en una unidad SEI NAL actual para

determinar un punto de funcionamiento aplicable a los mensajes SEI de anidado escalables encapsulados por la unidad SEI NAL actual.

5 **[0095]** Las técnicas de esta divulgación proporcionan un diseño que indica si un punto de funcionamiento aplicable a los mensajes SEI de anidado en una unidad SEI NAL es el punto de funcionamiento indicado por la información de identificación de capa en la cabecera de la unidad NAL de la unidad SEI NAL. La información de identificación de capa en la cabecera de la unidad NAL de una unidad SEI NAL puede incluir el valor de nuh_reserved_zero_6bits y el valor de nuh_temporal_id_plus1 de la cabecera de la unidad NAL. En otras palabras, las técnicas de esta divulgación pueden proporcionar un diseño para el uso de información de identificación de capa (por ejemplo, el valor de nuh_reserved_zero_6bits y nuh_temporal_id_plus1) en la cabecera de la unidad NAL de una unidad SEI NAL actual (es decir, la unidad SEI NAL que contiene el mensaje SEI de anidado escalable), a través de la indicación de si los mensajes SEI de anidado se aplican a un punto de funcionamiento por defecto identificado por la información de identificación de capa incluida en la cabecera de la unidad NAL de la unidad SEI NAL actual.

15 **[0096]** De esta manera, el codificador de vídeo 20 puede incluir, en un mensaje SEI de anidado escalable encapsulado por una unidad SEI NAL, un elemento sintáctico que indica si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a una sub-flujo de bits por defecto. El sub-flujo de bits por defecto puede ser una representación de punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL. Además, el codificador de vídeo 20 puede emitir un flujo de bits que incluye el mensaje SEI de anidado escalable.

25 **[0097]** De manera similar, un dispositivo, como descodificador de vídeo 30 u otro dispositivo, puede determinar, basándose al menos en parte en un elemento sintáctico en un mensaje SEI de anidado escalable encapsulado por una unidad SEI NAL, si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto. Como antes, el sub-flujo de bits por defecto puede ser una representación de punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL. Cuando el mensaje SEI de anidado es aplicable al sub-flujo de bits por defecto, el dispositivo puede usar el mensaje SEI de anidado en un funcionamiento en el sub-flujo de bits por defecto. Por ejemplo, el mensaje SEI de anidado puede incluir uno o más parámetros HRD. En este ejemplo, el dispositivo puede usar uno o más parámetros HRD para realizar una prueba de conformidad de flujo de bits que determina si el sub-flujo de bits por defecto cumple con una norma de codificación de vídeo, tal como HEVC. De forma alternativa, en este ejemplo, el dispositivo puede usar uno o más parámetros HRD para determinar si el descodificador de vídeo 30 satisface una prueba de conformidad del descodificador.

35 **[0098]** En otro ejemplo de los problemas o deficiencias de los procedimientos existentes para el anidado de los mensajes SEI, la codificación explícita de identificadores de capa es ineficiente. Las técnicas de esta divulgación pueden aumentar la eficiencia de la codificación explícita de identificadores de capa a través de codificación o codificación diferencial utilizando indicadores.

45 **[0099]** La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas de esta divulgación. La FIG. 2 se proporciona con fines de explicación y no debe considerarse limitante de las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta divulgación describe el codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

50 **[0100]** En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluyen una unidad de procesamiento de predicción 100, una unidad de generación residual 102, una unidad de procesamiento de transformada 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de procesamiento de transformada inversa 110, una unidad de reconstrucción 112, una unidad de filtro 114, una memoria intermedia de imágenes descodificadas 116, y una unidad de codificación por entropía 118. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de inter-predicción 120 y una unidad de procesamiento de intra-predicción 126. La unidad de procesamiento de inter-predicción 120 incluye una unidad de estimación de movimiento 122 y una unidad de compensación de movimiento 124. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales.

60 **[0101]** El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede codificar cada CTU en un fragmento de una imagen de los datos de vídeo. Cada una de las CTU puede asociarse con bloques de árbol de codificación (CTB) luma de igual tamaño y CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede realizar división de árbol cuádruple para dividir los CTB de la CTU en bloques progresivamente más pequeños. Los bloques más pequeños pueden ser bloques de codificación de CU. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un CTB asociado con una CTU en cuatro sub-bloques de igual tamaño, dividir uno o más de los sub-bloques en cuatro sub-sub-bloques de igual tamaño, etc.

65

[0102] El codificador de vídeo 20 puede codificar CU de un CTB para generar representaciones codificadas de las CU (es decir, CU codificadas). Como parte de la codificación de una CU, la unidad de procesamiento de predicción 100 puede dividir los bloques de codificación asociados con la CU entre una o más PU de la CU. Por lo tanto, cada PU puede asociarse con un bloque de predicción de luma y bloques de predicción de croma correspondientes. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden soportar PU de varios tamaños. Como se ha indicado anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU y el tamaño de una PU puede referirse al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular sea $2N \times 2N$, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden soportar tamaños de PU de $2N \times 2N$ o $N \times N$ para intra-predicción, y tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares para inter-predicción. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden soportar división asimétrica para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para inter-predicción.

[0103] La unidad de procesamiento de inter-predicción 120 puede generar datos predictivos para una PU realizando inter-predicción en cada PU de una CU. Los datos predictivos para la PU pueden incluir bloques predictivos de la PU e información de movimiento para la PU. La unidad de procesamiento de inter-predicción 120 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento en I, todas las PU son intra-predichas. Por lo tanto, si la PU está en un fragmento I, la unidad de procesamiento de inter-predicción 120 no realiza inter-predicción en la PU. Por lo tanto, para bloques codificados en el modo I, el bloque predictivo se forma usando predicción espacial a partir de bloques contiguos previamente codificados dentro de la misma trama.

[0104] Si una PU está en un fragmento P, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia en una lista de imágenes de referencia (por ejemplo, "RefPicList0") para una región de referencia para la PU. La región de referencia para la PU puede ser una región, dentro de una imagen de referencia, que contiene bloques de muestra que se corresponden más estrechamente con los bloques de muestra de la PU. La unidad de estimación de movimiento 122 puede generar un índice de referencia que indique una posición en RefPicList0 de la imagen de referencia que contiene la región de referencia para la PU. Además, la unidad de estimación de movimiento 122 puede generar un vector de movimiento que indica un desplazamiento espacial entre un bloque de codificación de la PU y una ubicación de referencia asociada con la región de referencia. Por ejemplo, el vector de movimiento puede ser un vector bidimensional que proporciona una desviación respecto a las coordenadas en la imagen actual a las coordenadas en una imagen de referencia. La unidad de estimación de movimiento 122 puede emitir el índice de referencia y el vector de movimiento como la información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar los bloques predictivos de la PU basándose en muestras reales o interpoladas en la ubicación de referencia indicada por el vector de movimiento de la PU.

[0105] Si una PU está en un fragmento B, la unidad de estimación de movimiento 122 puede llevar a cabo predicción unidireccional o predicción bidireccional para la PU. Para realizar predicción unidireccional para la PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de RefPicList0 o una segunda lista de imágenes de referencia ("RefPicList1") para una región de referencia para la PU. La unidad de estimación de movimiento 122 puede emitir, como la información de movimiento de la PU, un índice de referencia que indica una posición en RefPicList0 o RefPicList1 de la imagen de referencia que contiene la región de referencia, un vector de movimiento que indica un desplazamiento espacial entre un bloque de predicción de la PU y una ubicación de referencia asociada con la región de referencia, y uno o más indicadores de dirección de predicción que indican si la imagen de referencia está en RefPicList0 o RefPicList1. La unidad de compensación de movimiento 124 puede generar los bloques predictivos de la PU basándose al menos en parte en muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.

[0106] Para realizar inter-predicción bidireccional para una PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia en RefPicList0 para una región de referencia para la PU y también puede buscar las imágenes de referencia en RefPicList1 para otra región de referencia para la PU. La unidad de estimación de movimiento 122 puede generar índices de referencia que indiquen posiciones en RefPicList0 y RefPicList1 de las imágenes de referencia que contienen las regiones de referencia. Además, la unidad de estimación de movimiento 122 puede generar vectores de movimiento que indican desplazamientos espaciales entre las ubicaciones de referencia asociadas con las regiones de referencia y un bloque de predicción de la PU. La información de movimiento de la PU puede incluir los índices de referencia y los vectores de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar los bloques predictivos de la PU basándose al menos en parte en muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.

[0107] La unidad de procesamiento de intra-predicción 126 puede generar datos predictivos para una PU realizando intra-predicción en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos para la PU y varios elementos sintácticos. La unidad de procesamiento de intra-predicción 126 puede realizar la intra-predicción en PU en fragmentos I, fragmentos P y fragmentos B.

[0108] Para realizar la intra-predicción en una PU, la unidad de procesamiento de intra-predicción 126 puede usar múltiples modos de intra-predicción para generar múltiples conjuntos de datos predictivos para la PU. La unidad de procesamiento de intra-predicción 126 puede generar un bloque predictivo para una PU basándose en muestras de

5 PU contiguas. Las PU contiguas pueden estar arriba, arriba y a la derecha, arriba y a la izquierda, o a la izquierda de la PU, asumiendo un orden de codificación de arriba a abajo, de izquierda a derecha para las PU, CU y CTU. La unidad de procesamiento de intra-predicción 126 puede usar varios números de modos de intra-predicción, por ejemplo, 33 modos de intra-predicción direccional. En algunos ejemplos, el número de modos de intra-predicción puede depender del tamaño de los bloques de predicción de la PU.

10 **[0109]** La unidad de procesamiento de predicción 100 puede seleccionar los datos predictivos para PU de una CU entre los datos predictivos generados por la unidad de procesamiento de inter-predicción 120 para las PU o los datos predictivos generados por la unidad de procesamiento de de intra-predicción 126 para las PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos predictivos para la PU de la CU basándose en las métricas de velocidad / distorsión de los conjuntos de datos predictivos. Los bloques predictivos de los datos predictivos seleccionados pueden denominarse en el presente documento bloques predictivos seleccionados.

15 **[0110]** La unidad de generación residual 102 puede generar, basándose en los bloques de codificación de luma, Cb y Cr de una CU y los bloques de luma, Cb y Cr predictivos seleccionados de las PU de la CU, bloques residuales de luma, Cb y Cr de la CU. Por ejemplo, la unidad de generación residual 102 puede generar los bloques residuales de la CU de manera que cada muestra en los bloques residuales tenga un valor igual a una diferencia entre una muestra en un bloque de codificación de la CU y una muestra correspondiente en un bloque predictivo seleccionado correspondiente de una PU de la CU.

20 **[0111]** La unidad de procesamiento de transformada 104 puede realizar división de árbol cuádruple para dividir los bloques residuales de una CU en bloques de transformada asociados con TU de la CU. De este modo, una TU puede estar asociada con un bloque de transformada de luma y dos bloques de transformada de croma correspondientes. Los tamaños y posiciones de los bloques de transformada de luma y croma de TU de una CU pueden o no basarse en los tamaños y posiciones de los bloques de predicción de las PU de la CU. Una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT) puede incluir nodos asociados con cada una de las regiones. Las TU de una CU pueden corresponder a nodos de hoja del RQT.

25 **[0112]** La unidad de procesamiento de transformada 104 puede generar bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 104 puede aplicar varias transformadas a un bloque de transformada asociado con una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar al bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 104 no aplica transformadas a un bloque de transformada. En dichos ejemplos, el bloque de transformada puede tratarse como un bloque de coeficientes de transformada.

30 **[0113]** La unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o a la totalidad de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de n bits puede redondearse a un coeficiente de transformada de m bits durante la cuantificación, donde n es mayor que m . La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado con una TU de una CU basándose en un valor de parámetro de cuantificación (QP) asociado con la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados con una CU ajustando el valor QP asociado con la CU. La cuantificación puede introducir pérdida de información, por lo que los coeficientes de transformada cuantificados pueden tener una precisión inferior a los originales.

35 **[0114]** La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformada inversa 110 pueden aplicar cuantificación inversa y transformadas inversas a un bloque de coeficientes, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes. La unidad de reconstrucción 112 puede añadir el bloque residual reconstruido a las muestras correspondientes de uno o más bloques predictivos generados por la unidad de procesamiento de predicción 100 para producir un bloque de transformada reconstruido asociado con una TU. Mediante la reconstrucción de bloques de transformada para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

40 **[0115]** La unidad de filtro 114 puede realizar una o más operaciones de desbloqueo para reducir los artefactos de bloqueo en los bloques de codificación asociados con una CU. La memoria intermedia de imágenes descodificadas 116 puede almacenar los bloques de codificación reconstruidos después de que la unidad de filtro 114 realice la una o más operaciones de desbloqueo en los bloques de codificación reconstruidos. La unidad de procesamiento de inter-predicción 120 puede utilizar una imagen de referencia que contiene los bloques de codificación reconstruidos para realizar inter-predicción en las PU de otras imágenes. Además, la unidad de procesamiento de intra-predicción 126 puede usar bloques de codificación reconstruidos en la memoria intermedia de imágenes descodificadas 116 para realizar intra-predicción en otras PU en la misma imagen que la CU.

45 **[0116]** La unidad de codificación por entropía 118 puede recibir datos de otros componentes funcionales del

codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 118 puede recibir bloques de coeficientes de la unidad de cuantificación 106 y puede recibir elementos sintácticos de la unidad de procesamiento de predicción 100. La unidad de codificación por entropía 118 puede realizar una o más operaciones de codificación por entropía en los datos para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 118 puede realizar una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación de CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de división de intervalo de probabilidad (PIPE), una operación de codificación de Golomb exponencial u otro tipo de operación de codificación por entropía en los datos. El codificador de vídeo 20 puede emitir un flujo de bits que incluye datos codificados por entropía generados por la unidad de codificación por entropía 118. Por ejemplo, el flujo de bits puede incluir datos que representan un RQT para una CU.

[0117] Como se indicó anteriormente, las técnicas de esta divulgación pueden proporcionar un diseño para permitir que la información común de las estructuras sintácticas de parámetros HRD sea indicada explícitamente para cualquier estructura sintáctica de parámetros HRD en un VPS. Para permitir que la información común de estructuras sintácticas de parámetros HRD sea indicada explícitamente para cualquier parámetro HRD en un VPS, el codificador de vídeo 20 puede generar estructuras sintácticas VPS que se ajusten a la sintaxis de ejemplo mostrada en la Tabla 3 a continuación.

TABLA 3 - Estructura sintáctica VPS

	Descriptor
video_parameter_set_rbsp(){	
video_parameter_set_id	u(4)
vps_temporal_id_nesting_flag	u(1)
reserved_zero_2bits	u(2)
reserved_zero_6bits	u(6)
vps_max_sub_layers_minus1	u(3)
profile_tier_level(l, vps_max_sub_layers_minus1)	
next_essential_info_byte_offset //reserved_zero_12bits en la especificación base	
for(i = 0; i <= vps_max_sub_layers_minus1; i++) {	u(12)
vps_max_dec_pic_buffering[i]	ue(v)
vps_max_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
num_ops_minus1	ue(v)
for(i = 1; i <= num_ops_minus1; i++)	
operation_point(i)	
vps_num_hrd_parameters	ue(v)
for(i = 0; i < vps_num_hrd_parameters; i++) {	
hrd_applicable_ops_minus1[i]	ue(v)
for(j = 0; j <= hrd_applicable_ops_minus1[i]; j++){	
hrd_op_idx[i][j]	ue(v)
if(i > 0)	
cprms_present_flag[i]	
hrd_parameters(cprms_present_flag[i], vps_max_sub_layers_minus1)	
}	
vps_extension_flag	u(1)
si(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0118] Las partes en cursiva de la Tabla 3 indican diferencias entre la sintaxis de la Tabla 3 y la tabla correspondiente del Borrador de Trabajo 8 de HEVC. Además, en la sintaxis de ejemplo de la Tabla 3, el elemento sintáctico num_ops_minus1 especifica el número de estructuras sintácticas operation_point() presentes en el VPS. El elemento sintáctico hrd_applicable_ops_minus1[i] especifica el número de puntos de funcionamiento a los que se aplica la estructura sintáctica i-ésima hrd_parameters(). El elemento sintáctico hrd_op_idx[i][j] especifica el j-ésimo punto de funcionamiento al que se aplica la estructura sintáctica i-ésima hrd_parameters() en el VPS. Como se mencionó brevemente anteriormente, las técnicas de esta divulgación pueden permitir que un conjunto de parámetros HRD sea compartido por múltiples puntos de funcionamiento. Los elementos sintácticos hrd_applicable_ops_minus1[i] y hrd_op_idx[i][j] pueden servir para indicar los puntos de funcionamiento a los que se aplica un conjunto de parámetros HRD. En algunos ejemplos donde no se permite aplicar múltiples puntos de funcionamiento a un solo conjunto de parámetros HRD, los elementos sintácticos hrd_applicable_ops_minus1[i] y los

elementos sintácticos `hrd_op_idx[i][j]` se omiten de la Tabla 3.

[0119] En la sintaxis de ejemplo de la Tabla 3, un VPS puede incluir un conjunto de indicadores presentes de parámetros comunes (es decir, elementos sintácticos), denominados en la Tabla 3 `cprms_present_flag[i]`. Un elemento sintáctico `cprms_present_flag[i]` igual a 1 especifica que los parámetros HRD que son comunes para todas las sub-capas están presentes en la estructura sintáctica *i*-ésima `hrd_parameters()` en el VPS. Un elemento sintáctico `cprms_present_flag[i]` igual a 0 especifica que los parámetros HRD que son comunes para todas las sub-capas no están presentes en la estructura sintáctica *i*-ésima `hrd_parameters()` en el VPS y, por lo contrario, se obtienen para ser los mismos que la (*i*-1)-ésima estructura sintáctica `hrd_parameters()` en el VPS.

[0120] Se puede inferir que `cprms_present_flag[0]` es igual a 1. Es decir, un dispositivo puede determinar automáticamente (es decir, inferir) que la primera (en orden de codificación) estructura sintáctica `hrd_parameters()` en el VPS incluye parámetros HRD que son comunes para todas las sub-capas. En consecuencia, la primera estructura sintáctica de parámetros HRD indicada en el VPS incluye un conjunto común de parámetros HRD. Una o más estructuras sintácticas de parámetros HRD posteriores en el VPS pueden incluir diferentes conjuntos comunes de parámetros HRD.

[0121] Como se ha mencionado brevemente más arriba, las técnicas de esta divulgación pueden permitir que la información común (es decir, los parámetros HRD comunes a cada una de las sub-capas) de las estructuras sintácticas de parámetros HRD se indiquen explícitamente para cualquier estructura sintáctica de parámetros HRD. El elemento sintáctico `cprms_present_flag[i]` de la Tabla 3 puede permitir al descodificador de vídeo 30 u otro dispositivo determinar qué estructuras sintácticas de parámetros HRD incluyen un conjunto de parámetros HRD comunes a cada una de las sub-capas. Por lo tanto, aunque la primera estructura sintáctica de parámetros HRD puede incluir siempre el conjunto común de parámetros HRD, una o más estructuras sintácticas de parámetros HRD indicadas en el VPS no incluyen el conjunto común de parámetros HRD. Un dispositivo puede usar los elementos sintácticos `cprms_present_flag[i]` para determinar cuáles de las estructuras sintácticas de parámetros HRD del VPS incluyen conjuntos comunes de parámetros HRD.

[0122] Una estructura sintáctica de parámetros HRD (por ejemplo, una estructura sintáctica `hrd_parameters()`) puede incluir el conjunto de parámetros HRD específicos de sub-capa independientemente de si la estructura sintáctica de parámetros HRD incluye parámetros HRD que son comunes para todas las sub-capas. Cuando el descodificador de vídeo 30 u otro dispositivo determina que una estructura sintáctica de parámetros HRD particular no incluye el conjunto común de parámetros HRD, el descodificador 30 o el otro dispositivo puede realizar una operación usando un conjunto común de parámetros HRD asociados con una estructura sintáctica de parámetros HRD anterior y el conjunto de los parámetros HRD específicos de sub-capa de la estructura sintáctica de parámetros HRD particular. La estructura sintáctica de parámetros HRD anterior puede ser un conjunto de parámetros HRD indicados en el VPS antes, en orden de codificación, de la estructura sintáctica de parámetros HRD particular. Si la estructura sintáctica de parámetros HRD anterior incluye un conjunto común de parámetros HRD, el conjunto común de parámetros HRD asociados con la estructura sintáctica de parámetros HRD anterior es el conjunto común de parámetros HRD incluidos en la estructura sintáctica de parámetros HRD anterior. Si la estructura sintáctica de parámetros HRD anterior no incluye el conjunto común de parámetros HRD, un dispositivo puede determinar que el conjunto común de parámetros HRD asociados con la estructura sintáctica de parámetros HRD anterior es el conjunto común de parámetros HRD asociados con una estructura sintáctica de parámetros HRD anterior, en orden de codificación, a la estructura sintáctica de parámetros HRD anterior en orden de codificación.

[0123] Como se menciona anteriormente, un dispositivo puede realizar una operación usando un conjunto común de parámetros HRD y parámetros HRD específicos de sub-capa. Durante esta operación, el dispositivo puede gestionar el funcionamiento de un CPB de acuerdo con uno o más de los parámetros HRD, descodificar los datos de vídeo, y gestionar imágenes descodificadas en DPB de acuerdo con uno o más de los parámetros HRD. En otro ejemplo, el conjunto común de parámetros HRD y parámetros HRD específicos de sub-capa pueden usarse para realizar una prueba de conformidad de flujo de bits o una prueba de conformidad de descodificador.

[0124] Además, en algunos ejemplos, un mensaje SEI de anidado escalable proporciona un mecanismo para asociar mensajes SEI con subconjuntos de flujo de bits (por ejemplo, representaciones de punto de funcionamiento) o con capas y sub-capas específicas. En algunos ejemplos de este tipo, un mensaje SEI de anidado escalable puede contener uno o más mensajes SEI. Un mensaje SEI contenido en un mensaje SEI de anidado escalable puede denominarse mensaje SEI de anidado. Un mensaje SEI no contenido en un mensaje SEI de anidado escalable puede denominarse un mensaje SEI de no anidado. En algunos ejemplos, un mensaje SEI de anidado en un mensaje SEI de anidado escalable puede incluir un conjunto de parámetros HRD.

[0125] En algunos ejemplos, hay varias limitaciones sobre qué tipos de mensajes pueden anidarse. Por ejemplo, un mensaje SEI de período de almacenamiento en memoria intermedia y un mensaje SET de cualquier otro tipo pueden no anidarse en el mismo mensaje SEI de anidado escalable. Un mensaje SEI de período de almacenamiento en memoria intermedia puede indicar retardos iniciales para el funcionamiento de HRD. En otro ejemplo, un mensaje SEI de sincronización de imagen y un mensaje SEI de cualquier otro tipo pueden no anidarse en el mismo mensaje SEI de anidado escalable. Un mensaje SEI de sincronización de imagen puede indicar un tiempo de salida de

imagen y un tiempo de eliminación de imagen / sub-imagen para el funcionamiento de HRD. En otros ejemplos, un mensaje SEI de sincronización de imagen y un mensaje SEI de sincronización de sub-imagen pueden estar anidados en el mismo mensaje SEI de anidado escalable. Un mensaje SEI de sincronización de sub-imagen puede proporcionar información de retardo de eliminación de CPB para la unidad descodificada asociada con el mensaje SEI.

[0126] Como se indicó anteriormente, una o más técnicas de esta divulgación pueden permitir que un mensaje SEI se aplique a múltiples puntos de funcionamiento. Además, una o más técnicas de esta divulgación pueden permitir que el codificador de vídeo 20 indique si un punto de funcionamiento aplicable a un mensaje SEI de anidado en una unidad SEI NAL es el punto de funcionamiento indicado por la información de identificación de capa en la cabecera de la unidad NAL de la unidad SEI NAL. Además, una o más técnicas de esta divulgación pueden aumentar la eficiencia de la codificación explícita de identificadores de capa mediante codificación diferencial. La sintaxis de ejemplo que se muestra en la Tabla 4, a continuación, y la semántica que la acompaña pueden implementar estas técnicas.

TABLA 4 - Mensaje SEI de anidado escalable

	Descriptor
scalable_nesting(payloadSize) {	
bitstream_subset_flag	u(1)
if(<i>bitstream_subset_flag</i>) {	
default_op_applicable_flag	u(1)
nesting_num_ops_minus1	ue(v)
} else	
nesting_op_flag	u(1)
for(<i>i = 0; i <= nesting_num_ops_minus1; i++</i>)	
nesting_max_temporal_id_plus1[i]	u(3)
if(<i>nesting_op_flag</i>)	
for(<i>i = 0; i <= nesting_num_ops_minus1; i++</i>)	
nesting_op_idx[i]	ue(v)
else {	
all_layers_flag	u(1)
if(! <i>all_layers_flag</i>) {	
nesting_num_layers_minus1	ue(v)
for(<i>i = 0; i <= nesting_num_layers_minus1; i++</i>)	
nesting_layer_id_delta[i]	ue(v)
}	
}	
while(! <i>byte_aligned</i> ())	u(1)
nesting_zero_bit /*equal to 0*/	
Do	
sei_message()	
while(<i>more_rbsp_data</i> ())	
}	

[0127] En el ejemplo de la Tabla 4, las partes en cursiva pueden indicar diferencias de HEVC. Borrador de Trabajo 8. Específicamente, en la sintaxis de ejemplo de la Tabla 4, un elemento sintáctico *bitstream_subset_flag* igual a 0 especifica que los mensajes SEI de anidado en el mensaje SEI de anidado escalable se aplican a capas y sub-capas específicas. Un elemento sintáctico *bitstream_subset_flag* igual a 1 especifica que los mensajes SEI de anidado en el mensaje SEI de anidado escalable se aplican a un sub-flujo de bits resultante de un proceso de extracción de sub-flujo de bits de la subcláusula 10.1 del Borrador de Trabajo 8 de HEVC con entradas especificadas por los elementos sintácticos del mensaje SEI de anidado escalable como se especifica a continuación. La Subcláusula 10.1 del Borrador de Trabajo 8 de HEVC describe una operación para extraer un sub-flujo de bits (es decir, una representación de punto de funcionamiento) del flujo de bits. Específicamente, la subcláusula 10.1 del Borrador de Trabajo 8 de HEVC indica que el sub-flujo de bits se obtiene al eliminar del flujo de bits todas las unidades NAL con identificadores temporales (por ejemplo, *TemporalID*) mayores que *IdTarget* o identificadores de capa (por ejemplo, *nuh_reserved_zero_6bits*) que no estén entre los valores en *targetDecLayerIdSet*. *IdTarget* y *targetDecLayerIdSet* son parámetros del proceso de extracción de flujo de bits. En algunos ejemplos, si los mensajes SEI de anidado son mensajes SEI de almacenamiento en memoria intermedia de imágenes, mensajes SEI de sincronización de imágenes o mensajes SEI de sincronización de sub-imágenes, el elemento sintáctico *bitstream_subset_flag* es igual a 1. De lo contrario, en dichos ejemplos, el elemento sintáctico *bitstream_subset_flag* es igual a 0.

[0128] Además, en la sintaxis de ejemplo de la Tabla 4, el mensaje SEI de anidado escalable incluye un elemento sintáctico *default_op_applicable_flag* si el elemento sintáctico *bitstream_subset_flag* es igual a 1. Un elemento

sintáctico `default_op_applicable_flag` igual a 1 especifica que los mensajes SEI de anidado (es decir, los mensajes SEI de anidado dentro del mensaje SEI de anidado escalable) se aplican a un sub-flujo de bits por defecto que es el resultado del proceso de extracción sub-flujo de bits de la subcláusula 10.1 del Borrador de Trabajo 8 de HEVC con entradas `tdTarget` igual al identificador temporal (`TemporalId`) de la unidad SEI NAL actual y `targetDecLayerIdSet` que consta de todos los valores de `nuh_reserved_zero_6bits` en el rango de 0 a `nuh_reserved_zero_6bits` de la unidad SEI NAL actual, inclusive. Por lo tanto, el flujo de bits por defecto puede ser el flujo de bits que se obtiene al eliminar del flujo de bits todas las unidades NAL con identificadores temporales mayores que el identificador temporal de la unidad SEI NAL actual o los identificadores de capa en el rango de 0 al identificador de capa (por ejemplo, `nuh_reserved_zero_6bits`) de la unidad SEI NAL actual, inclusive. Por ejemplo, el sub-flujo de bits por defecto puede ser un subconjunto de un flujo de bits y el sub-flujo de bits por defecto puede no incluir unidades NAL VCL del flujo de bits que tienen identificadores de capa mayores que el identificador de capa indicado por el elemento sintáctico del identificador de capa de la cabecera de la unidad NAL o tener identificadores temporales mayores que el identificador temporal indicado por el elemento sintáctico del identificador de capa temporal (por ejemplo, `nuh_temporal_id_plus1`) de la cabecera de la unidad NAL. Un elemento sintáctico `default_op_applicable_flag` igual a 0 especifica que los mensajes SEI de anidado no se aplican al sub-flujo de bits por defecto.

[0129] En la sintaxis de ejemplo de la Tabla 4, el mensaje SEI de anidado escalable incluye un elemento sintáctico `nesting_num_ops_minus1` si el elemento sintáctico `bitstream_subset_flag` es igual a 1. El elemento sintáctico `nesting_num_ops_minus1`, más 1, especifica el número de elementos sintácticos `nesting_op_idx[i]` en el mensaje SEI de anidado escalable. Así, si el elemento sintáctico `nesting_num_ops_minus1`, más 1, es mayor que 0, el elemento sintáctico `nesting_num_ops_minus1` puede indicar si el mensaje SEI de anidado escalable incluye una pluralidad de elementos sintácticos que identifican múltiples puntos de funcionamiento para los cuales son aplicables los mensajes SEI de anidado. De esta forma, un dispositivo puede descodificar, a partir del mensaje SEI de anidado escalable, un elemento sintáctico (`nesting_num_ops_minus1`) que indica el número de puntos de funcionamiento para los cuales se aplica el mensaje SEI de anidado. Cuando el elemento sintáctico `nesting_num_ops_minus1` no está presente, puede inferirse que el valor de `nesting_num_ops_minus1` es igual a 0. Por lo tanto, si el elemento sintáctico `bitstream_subset_flag` es igual a 0, el mensaje SEI de anidado escalable no incluye elementos sintácticos `nesting_op_idx[i]`.

[0130] Un elemento sintáctico `nesting_op_flag` igual a 0 especifica que `nestingLayerIdSet[0]` es especificado por el elemento sintáctico `all_layers_flag` y, cuando está presente, un elemento sintáctico `nesting_layer_id_delta[i]` para todos los valores i en el rango de 0 a `nesting_num_layers_minus1`, inclusive. Los elementos sintácticos `nestingLayerIdSet[]` son una matriz de identificadores de capa. Un elemento sintáctico `nesting_op_flag` igual a 1 especifica que `nestingLayerIdSet[i]` está especificado por el elemento sintáctico `nesting_op_idx[i]`. Cuando no está presente, se infiere que el valor de `nesting_op_flag` es igual a 1.

[0131] El elemento sintáctico `nesting_max_temporal_id_plus1[i]` especifica una variable `maxTemporalId[i]`. En la sintaxis de ejemplo de la Tabla 4, el valor del elemento sintáctico `nesting_max_temporal_id_plus1[i]` es mayor que el valor del elemento sintáctico `nuh_temporal_id_plus1` de la unidad SEI NAL actual (es decir, la unidad NAL que contiene el mensaje SEI de anidado escalable). La variable `maxTemporalId[i]` se establece igual a `nesting_max_temporal_id_plus1[i] - 1`.

[0132] El elemento sintáctico `nesting_op_idx[i]` se usa para especificar el conjunto `nestingLayerIdSet[i]`. El conjunto `nestingLayerIdSet[i]` puede constar de `op_layer_id[nesting_op_idx[i]]` con todos los valores de en el rango de 0 a `op_num_layer_id_values_minus1[nesting_op_idx]`, inclusive. El VPS activo puede especificar los valores `op_layer_id[]` y `op_num_layer_values_minus1[]`.

[0133] Además, en la sintaxis de ejemplo de la Tabla 4, un elemento sintáctico `all_layers_flag` igual a 0 especifica que el conjunto `nestingLayerIdSet[0]` consta de `nestingLayerId[i]` para todos los valores i en el rango de 0 a `nesting_num_layers_minus1`, inclusive. La variable `nestingLayerId[i]` se describe a continuación. Un elemento sintáctico `all_layers_flag` igual a 1 especifica que el conjunto `nestingLayerIdSet` consta de todos los valores de `nuh_reserved_zero_6bits` presentes en la unidad de acceso actual que son iguales o mayores que `nuh_reserved_zero_6bits` de la unidad SEI NAL actual.

[0134] El elemento sintáctico `nesting_num_layers_minus1`, más 1, especifica el número de elementos sintácticos `nesting_layer_id_delta[i]` en el mensaje SEI de anidado escalable. Un elemento sintáctico `nesting_layer_id_delta[i]`, cuando i es igual a 0, especifica la diferencia entre el primer valor (es decir, el 0-ésimo) valor `nuh_reserved_zero_6bits` incluido en el conjunto `nestingLayerIdSet[0]` y el elemento sintáctico `nuh_reserved_zero_6bits` de la unidad SEI NAL actual. Un elemento sintáctico `nesting_layer_id_delta[i]`, cuando i es mayor que 0, especifica la diferencia entre los valores i -ésimo e $(i-1)$ -ésimo `nuh_reserved_zero_6bits` incluidos en el conjunto `nestingLayerIdSet[0]`.

[0135] La variable `nestingLayerId[i]` puede obtenerse de la siguiente manera, donde `nuh_reserved_zero_6bits` es de la cabecera de la unidad NAL de la unidad SEI NAL actual.

```

nestingLayerId[0]=nuh_reserved_zero_6bits+nesting_layer_id_delta[0]
for(i = 1; i<= nesting_num_layers_minus1; i++)

```

```

5         nestingLayerId[i]=nestingLayerId[i-1]nesting_layer_id_delta[i]

```

El conjunto nestingLayerIdSet[0] se configura para que conste de nestingLayerId [i] para todos los valores i en el rango de 0 a nesting_num_layers_minus1, inclusive. Cuando el elemento sintáctico bitstream_subset_flag es igual a 0, los mensajes SEI de anidado se aplican a unidades NAL con nuh_reserved_zero_6bits incluido en el conjunto nestingLayerIdSet [0] o igual a nuh_reserved_zero_6bits de la unidad SEI NAL actual, y con nuh_temporal_id_plus1 en el rango de nuh_temporal_id_plus1 de la unidad SEI NAL actual a maxTemporalId [0]+1, inclusive. Cuando el elemento sintáctico bitstream_subset_flag es igual a 1, los mensajes SEI de anidado se aplican a la salida del proceso de extracción de sub-flujo de bits de la subcláusula 10.1 del Borrador de Trabajo 8 de HEVC con entradas tldTarget iguales a maxTemporalId[i] y targetDecLayerIdSet igual a nestingLayerIdSet [i] para cada valor en el rango de 0 a nesting_num_ops_minus1, inclusive, y cuando el elemento sintáctico default_op_applicable_flag es igual a 1, los mensajes SEI de anidado también se aplican al sub-flujo de bits por defecto. El sub-flujo de bits extraído puede resultar de eliminar todas las unidades NAL con identificadores temporales mayores que maxTemporalId [i] o identificadores de capa en el rango de 0 a nesting_num_ops_minus1.

[0136] De esta manera, para al menos un respectivo punto de funcionamiento en la pluralidad de puntos de funcionamiento a los que se aplica el mensaje SEI de anidado, un dispositivo (por ejemplo, el codificador de vídeo 20, el decodificador de vídeo 30, u otro dispositivo, tal como un dispositivo de red de entrega de contenido) puede decodificar, a partir del mensaje SEI de anidado escalable, un primer elemento sintáctico (por ejemplo, nesting_max_temporal_id_plus1 [i]) y un segundo elemento sintáctico (por ejemplo, nesting_op_idx [i]). Además, el dispositivo puede determinar, basándose al menos en parte en el primer elemento sintáctico, un identificador temporal máximo del punto de funcionamiento respectivo. El dispositivo puede determinar, basándose al menos en parte en el segundo elemento sintáctico, un conjunto de identificadores de capa del punto de funcionamiento respectivo.

[0137] En el ejemplo de la Tabla 4, el elemento sintáctico nesting_zero_bit es igual a 0. El elemento sintáctico nesting_zero_bit puede servir para garantizar que el mensaje SEI de anidado escalable esté alineado por bytes. El mensaje SEI de anidado escalable puede estar alineado por bytes cuando el número de bits en el mensaje SEI de anidado escalable es divisible por 8.

[0138] Además, en el ejemplo de la Tabla 4, las estructuras sintácticas sei_message() incluyen mensajes SEI. De este modo, un dispositivo puede decodificar, a partir del mensaje SEI de anidado escalable, una pluralidad de mensajes SEI de anidado encapsulados por el mensaje SEI de anidado escalable. Cada uno de los mensajes SEI de anidado puede ser aplicable a todos los puntos de funcionamiento identificados por la pluralidad de elementos sintácticos (por ejemplo, nesting_max_temporal_id_plus1[i], nesting_op_idx[i], etc.).

[0139] En un ejemplo alternativo, los mensajes SEI de anidado escalables pueden ajustarse a la sintaxis de ejemplo de la Tabla 5 a continuación. En la sintaxis de ejemplo de la Tabla 5, un mensaje SEI de anidado escalable puede, de acuerdo con la una o más técnicas de esta divulgación, aumentar la eficiencia de la codificación explícita de identificadores de capa mediante el uso de indicadores de codificación.

TABLA 5 - Mensaje SEI de anidado escalable

scalable_nesting(payloadSize){	Descriptor
...	
if(bitstream_subset_flag){	
default_op_applicable_flag	u(l)
nesting_num_ops_minus1	ue(v)
} else	
nesting_op_flag	u(l)
for(i = 0; i <= nesting_num_ops_minus1; i++)	
nesting_max_temporal_id_plus1[i]	u(3)
if(nesting_op_flag)	
for(i = 0; i <= nesting_num_ops_minus1; i++)	
nesting_op_idx[i]	ue(v)
else{	
all_layers_flag	u(l)
if(!all_layers_flag){	
nesting_max_layer_id	u(6)
minLayerId = nuh_reserved_zero_6bits + 1	
for(i = 0; i < nesting_max_layer_id - minLayerId; i++)	

	Descriptor
scalable_nesting(payloadSize){	
<i>nesting_layer_id_included_flag[i]</i>	<i>ue(v)</i>
}	
}	
while(!byte_aligned())	
<i>nesting_zero_bit</i> /*equal two 0 */	<i>u(l)</i>
Do	
sei_message()	
while(more_rbsp_data())	
}	

[0140] En el ejemplo de la Tabla 5, las partes en cursiva muestran diferencias con el Borrador de Trabajo 8 de HEVC. Como se muestra en la Tabla 5, el elemento sintáctico *bitstream_subset_flag*, el elemento sintáctico *default_op_applicable_flag*, el elemento sintáctico *nesting_num_ops_minus1*, el elemento sintáctico *nesting_max_temporal_id_plus1*, el elemento sintáctico *nesting_op_idx [i]* y el elemento sintáctico *nesting_zero_bit* pueden tener la misma semántica que la descrita anteriormente con respecto a Tabla 4.

[0141] Además, en el ejemplo de la Tabla 5, una variable *minLayerId* se establece igual a *nuh_reserved_zero_6bits+1*, donde *nuh_reserved_zero_6bits* es de la cabecera de la unidad NAL de la unidad SEI NAL actual. Un elemento sintáctico *nesting_op_flag* igual a 0 especifica que el conjunto *nestingLayerIdSet[0]* es especificado por el elemento sintáctico *all_layers_flag* y, cuando está presente, *nesting_layer_id_included_flag [i]* para todos los valores *i* en el rango de 0 a *nesting_max_layer_id-minLayerId- 1*, inclusive. Un elemento sintáctico *nesting_op_flag* igual a 1 especifica que el conjunto *nestingLayerIdSet [i]* está especificado por el elemento sintáctico *nesting_op_idx [i]*. Cuando el elemento sintáctico *nesting_op_flag* no está presente, se infiere que el valor de *nesting_op_flag* es igual a 1.

[0142] En el ejemplo de la Tabla 5, un elemento sintáctico *all_layers_flag* igual a 0 especifica que el conjunto *nestingLayerIdSet[0]* consta de *nestingLayerId[i]* para todos los valores *i* en el rango de 0 a *nesting_max_layer_id-minLayerId*, inclusive. La variable *nestingLayerId[i]* se describe a continuación. En el ejemplo de la Tabla 5, un *all_layers_flag* igual a 1 especifica que el conjunto *nestingLayerIdSet* consta de todos los valores de *nuh_reserved_zero_6bits* presentes en la unidad de acceso actual que son mayores o iguales que el elemento sintáctico *nuh_reserved_zero_6bits* de la unidad SEI NAL actual.

[0143] Además, en el ejemplo de la Tabla 5, el elemento sintáctico *nesting_max_layer_id* especifica el mayor valor de *nuh_reserved_zero_6bits* en el conjunto *nestingLayerIdSet[0]*. Un elemento sintáctico *nesting_layer_id_included_flag[i]* igual a 1 especifica que el valor de *nuh_reserved_zero_6bits* es igual a *i + minLayerId* está incluido en el conjunto *nestingLayerIdSet[0]*. Un elemento sintáctico *nesting_layer_id_included_flag[i]*, igual a 0 especifica que el valor de *nuh_reserved_zero_6bits* igual a *i + minLayerId* no está incluido en el conjunto *nestingLayerIdSet[0]*.

[0144] La variable *nestingNumLayersMinus1* y las variables *nestingLayerId[i]* para *i* en el rango de 0 a *nestingNumLayersMinus1*, inclusive, pueden obtenerse de la siguiente manera:

```

for(i = 0, j = 0; i < nesting_max_layer_id; i++)
    if(nesting_layer_id_included_flag[i])
        nestingLayerId[j++] = i + minLayerId
nestingLayerId[j] = nesting_max_layer_id
nestingNumLayersMinus1 = j

```

El conjunto *nestingLayerIdSet[0]* se puede establecer para que consista en *nestingLayerId[i]* para todos los valores *i* en el rango de 0 a *nestingNumLayersMinus1*, inclusive.

[0145] Cuando el elemento sintáctico *bitstream_subset_flag* es igual a 0, los mensajes SEI de anidado pueden aplicarse a unidades NAL con *nuh_reserved_zero_6bits* incluidos en el conjunto *nestingLayerIdSet[0]* o igual al elemento sintáctico *nuh_reserved_zero_6bits* de la unidad actual SEI NAL, y con *nuh_temporal_id_plus* en el rango del elemento sintáctico *nuh_temporal_id_plus1* de la unidad actual SEI NAL a *maxTemporalId[0] + 1*, inclusive.

[0146] Cuando el elemento sintáctico *bitstream_subset_flag* del mensaje SEI de anidado escalable es igual a 1, los mensajes SEI de anidado pueden aplicarse a la salida del proceso de extracción de flujo secundario de la subcláusula 10.1 con entradas *tIdTarget* iguales a *maxTemporalId [i]* y *targetDecLayerIdSet* igual a *nestingLayerIdSet[i]* para cada valor *i* en el rango de 0 a *nesting_num_ops_minus1*, inclusive, y cuando

default_op_applicable_flag es igual a 1, los mensajes SEI de anidado también se aplican al sub-flujo de bits por defecto.

5 **[0147]** La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que está configurado para implementar las técnicas de esta divulgación. La FIG. 3 se proporciona con fines de explicación y no se limita a las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta divulgación describe el descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

10 **[0148]** En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 150, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de procesamiento de transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 160 y una memoria intermedia de imágenes descodificadas 162. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 164 y la unidad de procesamiento de intra-predicción 166. En otros
15 ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

[0149] Una memoria intermedia de imágenes codificadas (CPB) 151 puede recibir y almacenar datos de vídeo codificados (por ejemplo, unidades NAL) de un flujo de bits. La unidad de descodificación por entropía 150 puede recibir unidades NAL del CPB 151 y analizar las unidades NAL para descodificar elementos sintácticos. La unidad de descodificación por entropía 150 puede descodificar por entropía elementos sintácticos codificados por entropía en las unidades NAL. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de procesamiento de transformada inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 160 pueden generar datos de vídeo descodificados basándose en los elementos sintácticos extraídos del flujo de bits.
20

[0150] Las unidades NAL del flujo de bits pueden incluir unidades NAL de fragmentos codificados. Como parte de la descodificación del flujo de bits, la unidad de descodificación por entropía 150 puede extraer y descodificar por entropía elementos sintácticos de las unidades de NAL de fragmentos codificados. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos sintácticos pertenecientes a un fragmento. Los elementos sintácticos en la cabecera de fragmento pueden incluir un elemento sintáctico que identifica un PPS asociado con una imagen que contiene el fragmento.
25
30

[0151] Además de los elementos sintácticos de descodificación del flujo de bits, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Realizando la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir bloques residuales de la CU.
35

[0152] Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede cuantificar de forma inversa, es decir, descuantificar, los bloques de coeficientes asociados con la TU. La unidad de cuantificación inversa 154 puede utilizar un valor QP asociado con la CU de la TU para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para la unidad de cuantificación inversa 154 a aplicar. Es decir, la relación de compresión, es decir, la relación del número de bits utilizados para representar la secuencia original y la comprimida, puede controlarse ajustando el valor de la QP utilizada al cuantificar los coeficientes de transformada. La relación de compresión también puede depender del procedimiento de codificación por entropía empleado.
40
45

[0153] Después de que la unidad de cuantificación inversa 154 cuantifique inversamente un bloque de coeficientes, la unidad de procesamiento de transformada inversa 156 puede aplicar una o más transformadas inversas al bloque de coeficientes para generar un bloque residual asociado con la TU. Por ejemplo, la unidad de procesamiento de transformada inversa 156 puede aplicar una DCT inversa, una transformada de número entero inversa, una transformada de Karhunen-Loeve (KLT) inversa, una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes.
50

[0154] Si se codifica una PU usando intra-predicción, la unidad de procesamiento de intra-predicción 166 puede realizar la intra-predicción para generar bloques predictivos para la PU. La unidad de procesamiento de intra-predicción 166 puede usar un modo de intra-predicción para generar los bloques predictivos de luma, Cb y Cr para la PU basándose en los bloques de predicción de PU espacialmente contiguas. La unidad de procesamiento de intra-predicción 166 puede determinar el modo de intra-predicción para la PU basándose en uno o más elementos sintácticos descodificados a partir del flujo de bits.
55
60

[0155] La unidad de procesamiento de predicción 152 puede construir una primera lista de imágenes de referencia (RefPicList0) y una segunda lista de imágenes de referencia (RefPicList1) basándose en los elementos sintácticos extraídos del flujo de bits. Además, si se codifica una PU usando inter-predicción, la unidad de descodificación por entropía 150 puede extraer información de movimiento para la PU. La unidad de compensación de movimiento 164 puede determinar, basándose en la información de movimiento de la PU, una o más regiones de referencia para la
65

PU. La unidad de compensación de movimiento 164 puede generar, basándose en bloques de muestras en uno o más bloques de referencia para la PU, bloques predictivos de luma, Cb y Cr para la PU.

[0156] La unidad de reconstrucción 158 puede utilizar los bloques de transformada de luma, Cb y Cr asociados con las TU de una CU y los bloques de luma, Cb y Cr predictivos de las PU de la CU, es decir, datos de intra-predicción o datos de inter-predicción, según corresponda, para reconstruir los bloques de codificación de luma, Cb y Cr de la CU. Por ejemplo, la unidad de reconstrucción 158 puede agregar muestras de los bloques de transformada de luma, Cb y Cr a las muestras correspondientes de los bloques de luma, Cb y Cr predictivos para reconstruir los bloques de codificación de luma, Cb y Cr de la CU.

[0157] La unidad de filtro 160 puede 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 descodificador de vídeo 30 puede almacenar los bloques de codificación de luma, Cb y Cr de la CU en la memoria intermedia de imágenes descodificadas 162. La memoria intermedia de imágenes descodificadas 162 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intra-predicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en los bloques de luma, Cb y Cr en la memoria intermedia de imágenes descodificadas 162, operaciones de intra-predicción o de inter-predicción en PU de otras CU. De esta forma, el descodificador de vídeo 30 puede extraer, a partir del flujo de bits, los niveles de coeficientes de transformada del bloque de coeficientes luma significativos, cuantificar inversamente los niveles de coeficientes de transformada, aplicar una transformada a los niveles de coeficientes de transformada para generar un bloque de transformada, generar, basándose al menos en parte en el bloque de transformada, un bloque de codificación, y emitir el bloque de codificación para visualizar.

[0158] La FIG. 4 es un diagrama de flujo que ilustra un ejemplo de funcionamiento 200 de un codificador de vídeo 20 de acuerdo con una o más técnicas de esta divulgación. En el ejemplo de la FIG. 4, el codificador de vídeo 20 puede generar un VPS que incluye una pluralidad de estructuras sintácticas de parámetros HRD, cada una de las cuales incluye parámetros HRD (202). Para cada estructura sintáctica de parámetros HRD respectiva en la pluralidad de estructuras sintácticas de parámetros HRD, el VPS incluye además un elemento sintáctico que indica si entre los parámetros de HRD de la estructura sintáctica de parámetros HRD respectiva se incluye un conjunto común de parámetros HRD además de un conjunto de información de parámetros HRD específicos de sub-capa específico para una sub-capa particular del flujo de bits, en el que el conjunto común de parámetros HRD es común a todas las sub-capas del flujo de bits. Además, el codificador de vídeo 20 puede indicar el VPS en el flujo de bits (204).

[0159] La FIG. 5 es un diagrama de flujo que ilustra un ejemplo de funcionamiento 250 de un dispositivo, de acuerdo con una o más técnicas de esta divulgación. La operación 250 puede realizarse mediante el codificador de vídeo 20, el descodificador de vídeo 30 u otro dispositivo. Como se ilustra en el ejemplo de la FIG. 5, el dispositivo puede descodificar, a partir de un flujo de bits, un VPS que incluye una pluralidad de estructuras sintácticas de parámetros HRD, cada una de las cuales incluye parámetros HRD (252). Para cada estructura sintáctica de parámetros HRD respectiva en la pluralidad de estructuras sintácticas de parámetros HRD, el VPS incluye además un elemento sintáctico que indica si los parámetros HRD de la estructura sintáctica de parámetros HRD respectiva incluyen un conjunto común de parámetros HRD.

[0160] Además, el dispositivo puede realizar una operación usando los parámetros HRD de al menos una de las estructuras sintácticas de parámetros HRD (254). En algunos ejemplos, el flujo de bits puede comprender una representación de punto de funcionamiento de un punto de funcionamiento particular, una estructura sintáctica de parámetros HRD particular puede ser aplicable al punto de funcionamiento particular, y el dispositivo puede realizar la operación usando los parámetros HRD de la estructura sintáctica de parámetros HRD particular. Por ejemplo, el dispositivo puede usar los parámetros HRD para realizar una prueba de conformidad de flujo de bits que determina si un punto de funcionamiento aplicable a la estructura sintáctica de parámetros HRD se ajusta a una norma de codificación de vídeo, tal como HEVC. En otro ejemplo, el dispositivo puede usar los parámetros HRD para realizar una prueba de conformidad del descodificador.

[0161] El conjunto común de parámetros HRD puede ser común a todas las sub-capas del flujo de bits. En algunos ejemplos, entre los parámetros HRD de cada estructura sintáctica de parámetros HRD se incluye un conjunto de parámetros HRD específicos de sub-capa que es específico de una sub-capa particular del flujo de bits. En algunos ejemplos, cada uno de los conjuntos de parámetros HRD específicos de sub-capa incluye un elemento sintáctico (por ejemplo, indica una distancia temporal entre los tiempos de salida de HRD de dos imágenes consecutivas cualesquiera en orden de salida, un elemento sintáctico que indica una cantidad de especificaciones de memoria intermedia de imágenes codificadas alternativas en el flujo de bits de una secuencia de vídeo codificada. En algunos ejemplos, cuando el dispositivo determina que una estructura sintáctica de parámetros HRD particular no incluye un conjunto común de parámetros HRD, el dispositivo puede realizar la operación usando un conjunto común de parámetros HRD asociados con una estructura sintáctica de parámetros HRD anterior y el conjunto de parámetros HRD específicos de sub-capa de las estructuras sintácticas de parámetros HRD particulares.

[0162] La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de funcionamiento 300 de un codificador de vídeo 20

de acuerdo con una o más técnicas de esta divulgación. Como se ilustra en el ejemplo de la FIG. 6, el codificador de vídeo 20 puede generar un mensaje SEI de anidado escalable que incluye una pluralidad de elementos sintácticos que identifican una pluralidad de puntos de funcionamiento a los que se aplica un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable (302). Además, el codificador de vídeo 20 puede indicar el mensaje SEI de anidado escalable en un flujo de bits (304).

[0163] La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de funcionamiento 350 de un dispositivo, de acuerdo con una o más técnicas de esta divulgación. El codificador de vídeo 20, el descodificador de vídeo 30 u otro dispositivo pueden realizar la operación 350. Como se ilustra en el ejemplo de la FIG. 7, un dispositivo puede descodificar, a partir de un mensaje SEI de anidado escalable, una pluralidad de elementos sintácticos que identifican una pluralidad de puntos de funcionamiento a los que se aplica un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable (352). En algunos ejemplos, el dispositivo puede descodificar, a partir del mensaje SEI de anidado escalable, un elemento sintáctico (por ejemplo, `nesting_num_ops_minus1`) que indica si el mensaje `nesting_SEI` escalable incluye la pluralidad de elementos sintácticos que identifican los puntos de funcionamiento

[0164] Además, el dispositivo puede usar uno o más elementos sintácticos del mensaje SEI de anidado para realizar una operación con respecto a cualquiera de los puntos de funcionamiento a los que se aplica el mensaje SEI de anidado (354). Por ejemplo, el dispositivo puede usar elementos sintácticos del mensaje SEI de anidado en una prueba de conformidad de flujo de bits que determina si alguno de los puntos de funcionamiento a los que se aplica el mensaje SEI de anidado se ajusta a una norma de codificación de vídeo, como HEVC. En otro ejemplo, el dispositivo puede usar elementos sintácticos del mensaje SEI de anidado para realizar una prueba de conformidad del descodificador.

[0165] La FIG. 8 es un diagrama de flujo que ilustra un ejemplo de funcionamiento 400 de un codificador de vídeo 20 de acuerdo con una o más técnicas de esta divulgación. Como se ilustra en el ejemplo de la FIG. 8, el codificador de vídeo 20 puede incluir, en un mensaje SEI de anidado escalable encapsulado por una unidad SEI NAL, un elemento sintáctico (por ejemplo, `default_op_applicable_flag`) que indica si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto (402). El sub-flujo de bits por defecto es una representación de un punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL. Un primer elemento sintáctico en la cabecera de la unidad NAL (por ejemplo, `nuh_reserved_zero_6bits`) puede indicar el identificador de capa y un segundo elemento sintáctico en la cabecera de la unidad NAL (por ejemplo, `nuh_reserved_temporal_id_plus1`) puede indicar el identificador temporal.

[0166] En el ejemplo de la FIG. 8, el codificador de vídeo 20 puede incluir, en el mensaje SEI de anidado escalable, uno o más elementos sintácticos adicionales que identifican un identificador temporal de un punto de funcionamiento adicional y un identificador de capa máximo del punto de funcionamiento adicional (404). Además, el codificador de vídeo 20 puede indicar el mensaje SEI de anidado escalable en un flujo de bits (406). En algunos ejemplos, el elemento sintáctico que indica si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable al sub-flujo de bits por defecto se puede denominar un primer elemento sintáctico y el codificador de vídeo 20 puede incluir un segundo elemento sintáctico (por ejemplo, `bitstream_subset_flag`) en el mensaje SEI de anidado escalable. El segundo elemento sintáctico puede indicar si los mensajes SEI de anidado encapsulados por el mensaje SEI de anidado escalable se aplican a un sub-flujo de bits extraído del flujo de bits o si los mensajes SEI de anidado se aplican a capas y sub-capas específicas del flujo de bits. El codificador de vídeo 20 puede incluir el primer elemento sintáctico solo cuando el segundo elemento sintáctico indique que los mensajes SEI de anidado se aplican al sub-flujo de bits extraído del flujo de bits.

[0167] La FIG. 9 es un diagrama de flujo que ilustra un ejemplo de funcionamiento 450 de un dispositivo, de acuerdo con una o más técnicas de esta divulgación. El codificador de vídeo 20, el descodificador de vídeo 30 u otro dispositivo pueden realizar la operación 450. Como se ilustra en el ejemplo de la FIG. 9, el dispositivo puede determinar, basándose al menos en parte en un primer elemento sintáctico (por ejemplo, `bitstream_subset_flag`) de un mensaje `nesting_SEI` escalable, si un mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable se aplica a un sub-flujo de bits extraído de un flujo de bits (452). En respuesta a la determinación de que el mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable se aplica a un sub-flujo de bits extraído de un flujo de bits ("Sí" de 452), el dispositivo puede descodificar un elemento sintáctico de punto de funcionamiento por defecto (por ejemplo, `default_op_applicable_flag`) en el mensaje SEI de anidado escalable (454). El elemento sintáctico de punto de funcionamiento por defecto puede indicar si el mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto.

[0168] El sub-flujo de bits por defecto puede ser una representación de punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL y un identificador temporal especificado en la cabecera de la unidad NAL. En algunos ejemplos, un primer elemento sintáctico en la cabecera de la unidad NAL (por ejemplo, `nuh_reserved_zero_6bits`) indica el identificador de capa y un segundo elemento sintáctico en la cabecera de la unidad NAL (por ejemplo,

nuh_reserved_temporal_id_plus1) indica el identificador temporal. El sub-flujo de bits por defecto puede ser un subconjunto de un flujo de bits y el sub-flujo de bits por defecto no incluye unidades NAL VCL del flujo de bits que tienen identificadores de capa mayores que el identificador de capa indicado por el primer elemento sintáctico de la cabecera de la unidad NAL o tienen identificadores temporales mayores que el identificador temporal indicado por el segundo elemento sintáctico de la cabecera de la unidad NAL.

[0169] Además, el dispositivo puede determinar, basándose al menos en parte en un elemento sintáctico (por ejemplo, default_op_applicable_flag) en el mensaje SEI de anidado escalable encapsulado por la unidad SEI NAL, si el mensaje SEI de anidado encapsulado por el mensaje SEI de anidado escalable es aplicable a un sub-flujo de bits por defecto del flujo de bits (456). En algunos ejemplos, el mensaje SEI de anidado escalable encapsula una pluralidad de mensajes SEI de anidado. En dichos ejemplos, el dispositivo puede determinar, basándose al elemento sintáctico (por ejemplo, default_op_applicable_flag), si cada uno de los mensajes SEI de anidado en el mensaje SEI de anidado escalable es aplicable al sub-flujo de bits por defecto.

[0170] Cuando el mensaje SEI de anidado es aplicable al sub-flujo de bits por defecto ("SÍ" de 456), el dispositivo puede usar el mensaje SEI de anidado en una operación en el sub-flujo de bits por defecto (458). Por ejemplo, el mensaje SEI de anidado puede incluir un conjunto de parámetros HRD. En este ejemplo, el dispositivo puede usar los parámetros HRD en el mensaje SEI de anidado en una operación que prueba si el sub-flujo de bits por defecto se ajusta a una norma de codificación de vídeo, tal como HEVC. En otro ejemplo, el dispositivo puede usar los parámetros HRD en el mensaje SEI de anidado en una prueba de conformidad del descodificador. En otro ejemplo, el dispositivo puede usar el mensaje SEI de anidado en una operación de descodificación en el sub-flujo de bits por defecto. En otro ejemplo, los retardos iniciales en la eliminación de CPB pueden usarse para guiar a un sistema para establecer un retardo inicial apropiado de extremo a extremo y los tiempos de salida DPB pueden usarse para obtener las marcas de tiempo RTP cuando el vídeo se transporta sobre RTP.

[0171] De lo contrario, cuando el mensaje SEI de anidado no es aplicable al sub-flujo de bits por defecto ("NO" de 456) o cuando el mensaje SEI de anidado escalable no se aplica a un sub-flujo de bits extraído del flujo de bits ("NO" de 452), el dispositivo no utiliza el mensaje SEI de anidado en un funcionamiento en el sub-flujo de bits por defecto (460). Por ejemplo, el dispositivo puede determinar, basándose en uno o más elementos sintácticos adicionales en el mensaje SEI de anidado escalable (por ejemplo, nesting_max_temporal_id_plus1[i], nesting_op_idx[i], etc.) un identificador temporal de un segundo punto de funcionamiento y un identificador de capa máxima del segundo punto de funcionamiento. En este ejemplo, el dispositivo puede usar el mensaje SEI de anidado en una operación en un sub-flujo de bits adicional, siendo el sub-flujo de bits adicional una representación de punto de funcionamiento del segundo punto de funcionamiento.

[0172] En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse, como una o más instrucciones o código, en un medio legible por ordenador o transmitirse a través de este, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos o unos medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) unos medios de almacenamiento tangibles legibles por ordenador que son no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0173] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda utilizarse para almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, tal como se utiliza en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, donde algunos discos habitualmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior deberían incluirse también dentro del alcance de los medios legibles por ordenador.

5 **[0174]** Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables in situ (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquier estructura anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y la descodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

10 **[0175]** Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por medio de un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como los descritos anteriormente, conjuntamente con software y/o firmware adecuados.

10

15

REIVINDICACIONES

1. Un procedimiento de procesamiento de datos de vídeo, comprendiendo el procedimiento:
- 5 determinar si uno o más mensajes SEI de información de mejora suplementaria de anidado encapsulados por un mensaje SEI de anidado escalable son aplicables a un sub-flujo de bits por defecto de un flujo de bits de vídeo codificado, siendo el sub-flujo de bits por defecto una representación de punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa y un identificador temporal, estando el identificador de capa y el identificador temporal especificados en una cabecera de la
- 10 unidad de capa de abstracción de red, NAL, de una unidad SEI NAL que encapsula el mensaje SEI de anidado escalable; y
- cuando el uno o más mensajes SEI de anidado son aplicables al sub-flujo de bits por defecto, usar el uno o más mensajes SEI de anidado en un funcionamiento en el sub-flujo de bits por defecto,
- 15 en el que la determinación de si el uno o más mensajes SEI de anidado encapsulados por los mensajes SEI de anidado escalables son aplicables al sub-flujo de bits por defecto se basa al menos en parte en un elemento sintáctico en el mensaje SEI de anidado escalable, en el que el elemento sintáctico no está en ninguno de los mensajes SEI de anidado.
- 20 2. El procedimiento según la reivindicación 1, en el que el uso del mensaje SEI de anidado comprende el uso del mensaje SEI de anidado en una operación de descodificación en el sub-flujo de bits por defecto.
3. El procedimiento según la reivindicación 1, en el que:
- 25 el mensaje SEI de anidado escalable encapsula una pluralidad de mensajes SEI de anidado, y determinar si el mensaje SEI de anidado es aplicable al sub-flujo de bits por defecto comprende determinar, basándose al menos en parte en el elemento sintáctico, si cada uno de los mensajes SEI de anidado es aplicable al sub-flujo de bits por defecto.
- 30 4. El procedimiento según la reivindicación 1, en el que:
- el elemento sintáctico en el mensaje SEI de anidado escalable es un primer elemento sintáctico, y el procedimiento comprende además determinar, basándose al menos en parte en un segundo elemento sintáctico en el mensaje SEI de anidado escalable que indica que el uno o más mensajes SEI de anidado encapsulados por el mensaje SEI se aplican a un sub-flujo de bits extraído del flujo de bits de vídeo codificado, que el mensaje SEI de anidado escalable incluye el primer elemento sintáctico.
- 35 5. El procedimiento según la reivindicación 1, en el que el procedimiento comprende además:
- 40 determinar, basándose en uno o más elementos sintácticos adicionales en el mensaje SEI de anidado escalable, un identificador temporal de un segundo punto de funcionamiento y un identificador de capa máximo del segundo punto de funcionamiento; y
- 45 usar el mensaje SEI de anidado en una operación en un sub-flujo de bits adicional, siendo el sub-flujo de bits adicional una representación de punto de funcionamiento del segundo punto de funcionamiento.
6. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:
- 50 indicar un mensaje SEI de información de mejora suplementaria de anidado escalable en un flujo de bits de vídeo codificado, estando el mensaje SEI de anidado escalable en una unidad NAL de capa de abstracción de red SEI, y encapsular uno o más mensajes SEI de anidado,
- el procedimiento **caracterizado por** incluir, en el mensaje SEI de anidado escalable, un elemento sintáctico que indica si el uno o más mensajes SEI de anidado son aplicables a un sub-flujo de bits por defecto del flujo de bits de vídeo codificado, en el que el elemento sintáctico no está en ninguno de los
- 55 mensajes SEI de anidado y el sub-flujo de bits por defecto es una representación del punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en una cabecera de la unidad NAL de la unidad SEI NAL que encapsula el mensaje SEI de anidado escalable y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL que encapsula el mensaje SEI de anidado escalable.
- 60 7. El procedimiento según la reivindicación 1 o la reivindicación 6, en el que un primer elemento sintáctico en la cabecera de la unidad NAL indica el identificador de capa y un segundo elemento sintáctico en la cabecera de la unidad NAL indica el identificador temporal.
- 65 8. El procedimiento según la reivindicación 7, en el que el sub-flujo de bits por defecto es un subconjunto del flujo de bits de vídeo codificado y el sub-flujo de bits por defecto no incluye unidades NAL de capa de

codificación de vídeo (VCL) del flujo de bits de vídeo codificado que tienen identificadores de capa mayores que el identificador de capa indicado por el primer elemento sintáctico en la cabecera de la unidad NAL o tienen identificadores temporales mayores que el identificador temporal indicado por el segundo elemento sintáctico en la cabecera de la unidad NAL.

- 5
9. El procedimiento según la reivindicación 1 o la reivindicación 6, en el que el mensaje SEI de anidado incluye un conjunto de parámetros de descodificador hipotético de referencia, HRD.
- 10
10. El procedimiento según la reivindicación 9, que comprende además usar los parámetros HRD en el mensaje SEI de anidado en una operación que prueba si el sub-flujo de bits por defecto se ajusta a una norma de codificación de vídeo.
11. El procedimiento según la reivindicación 6, en el que:
- 15
- el mensaje SEI de anidado escalable encapsula una pluralidad de mensajes SEI de anidado, y el elemento sintáctico indica si cada uno de los mensajes SEI de anidado es aplicable al sub-flujo de bits por defecto.
12. El procedimiento según la reivindicación 6, en el que:
- 20
- el elemento sintáctico en el mensaje SEI de anidado escalable es un primer elemento sintáctico en el mensaje SEI, el procedimiento comprende además incluir un segundo elemento sintáctico en el mensaje SEI de anidado escalable, indicando el segundo elemento sintáctico si los mensajes SEI de anidado encapsulados por el mensaje SEI se aplican a un sub-flujo de bits extraído del flujo de bits de vídeo codificado o si los mensajes SEI de anidado se aplican a capas y sub-capas específicas del flujo de bits de vídeo codificado, y
- 25
- el mensaje SEI de anidado escalable incluye el primer elemento sintáctico solo cuando el segundo elemento sintáctico indica que los mensajes SEI de anidado se aplican al sub-flujo de bits extraído del flujo de bits de vídeo codificado.
- 30
13. El procedimiento según la reivindicación 6, en el que:
- 35
- el punto de funcionamiento identificado por uno o más elementos sintácticos de una cabecera de la unidad NAL de la unidad SEI NAL es un primer punto de funcionamiento, y el procedimiento comprende además incluir, en el mensaje SEI de anidado escalable, uno o más elementos sintácticos adicionales que identifican un identificador temporal de un segundo punto de funcionamiento y un identificador de capa máximo del segundo punto de funcionamiento.
- 40
14. Un dispositivo de procesamiento de vídeo, que comprende:
- 45
- medios para determinar si uno o más mensajes SEI de información de mejora suplementaria de anidado encapsulados por un mensaje SEI de anidado escalable son aplicables a un sub-flujo por defecto de un flujo de bits de vídeo codificado, siendo el sub-flujo de bits por defecto una representación de punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa y un identificador temporal, el identificador de capa y el identificador temporal se especifican en una capa de abstracción de red, NAL, cabecera de la unidad de una unidad SEI NAL que encapsula el mensaje SEI de anidado escalable;
- 50
- medios para usar, cuando el uno o más mensajes SEI de anidado son aplicables al sub-flujo de bits por defecto, los mensajes SEI de anidado en un funcionamiento en el sub-flujo de bits por defecto, en el que la determinación de si el uno o más mensajes SEI de anidado encapsulados por los mensajes SEI de anidado escalables son aplicables al sub-flujo de bits por defecto se basa al menos en parte en un elemento sintáctico en el mensaje SEI de anidado escalable, en el que el elemento sintáctico no está en ninguno de los mensajes SEI de anidado.
- 55
15. Un dispositivo de codificación de vídeo que comprende:
- 60
- medios para indicar un mensaje SEI de información de mejora suplementaria de anidado escalable en un flujo de bits de vídeo codificado, estando el mensaje SEI de anidado escalable encapsulado en una unidad de capa de abstracción de red, NAL, SEI y encapsular uno o más mensajes SEI de anidado, el dispositivo de codificación de vídeo **caracterizado por** medios para incluir, en el SEI de anidado escalable, un elemento sintáctico que indica si el uno o más mensajes SEI de anidado son aplicables a un sub-flujo por defecto de un flujo de bits de vídeo codificado, en el que el elemento sintáctico no está en ninguno de los mensajes SEI de anidado y el sub-flujo de bits por defecto es una representación de punto de funcionamiento de un punto de funcionamiento definido por un identificador de capa especificado en
- 65

una cabecera de la unidad NAL de la unidad SEI NAL que encapsula el mensaje SEI de anidado escalable y un identificador temporal especificado en la cabecera de la unidad NAL de la unidad SEI NAL que encapsula el mensaje SEI de anidado escalable.

- 5 **16.** Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando son ejecutadas por un dispositivo de codificación de vídeo, configuran el dispositivo de vídeo para realizar el procedimiento de acuerdo con una cualquiera de las reivindicaciones 1 a 13.

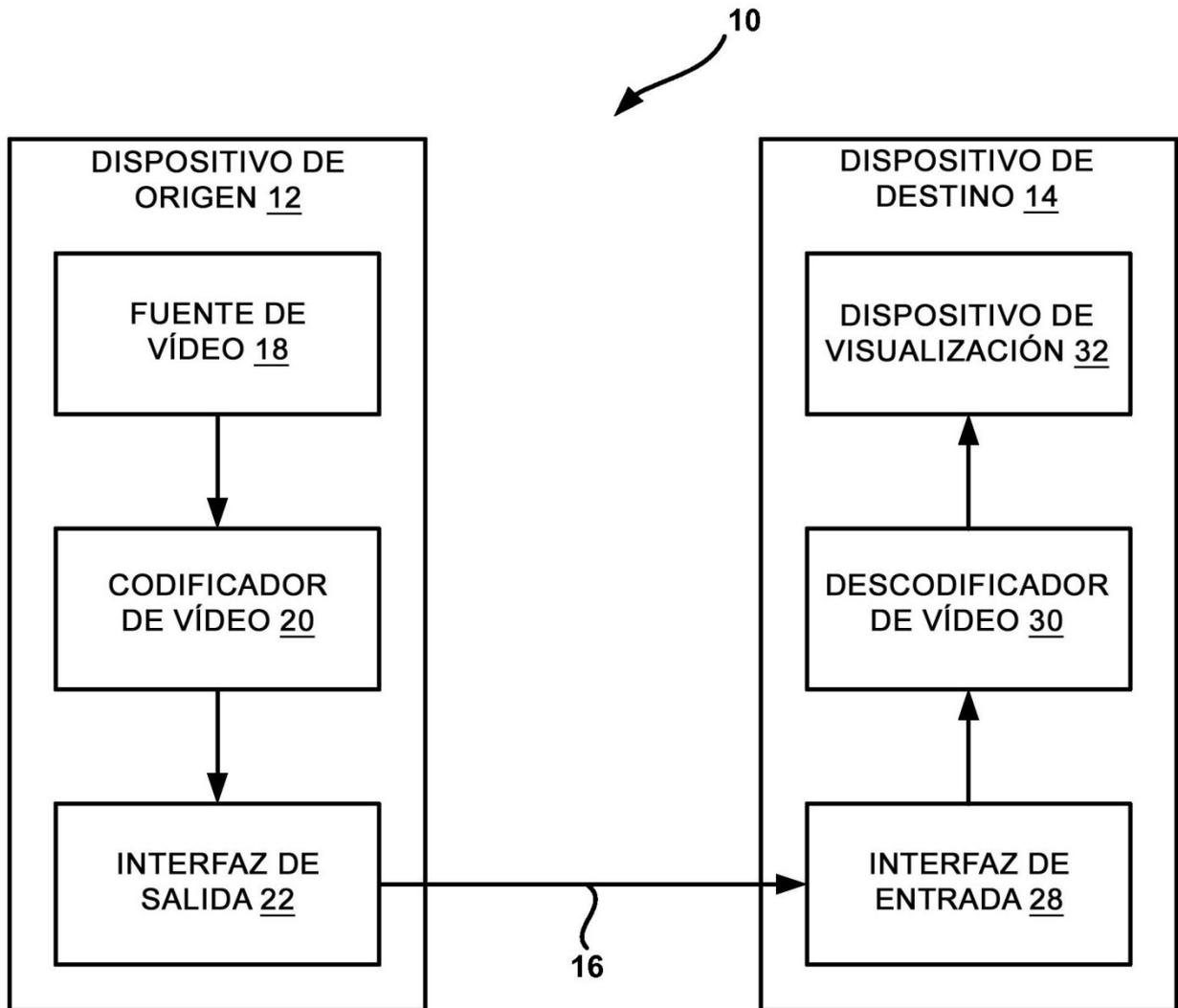


FIG. 1

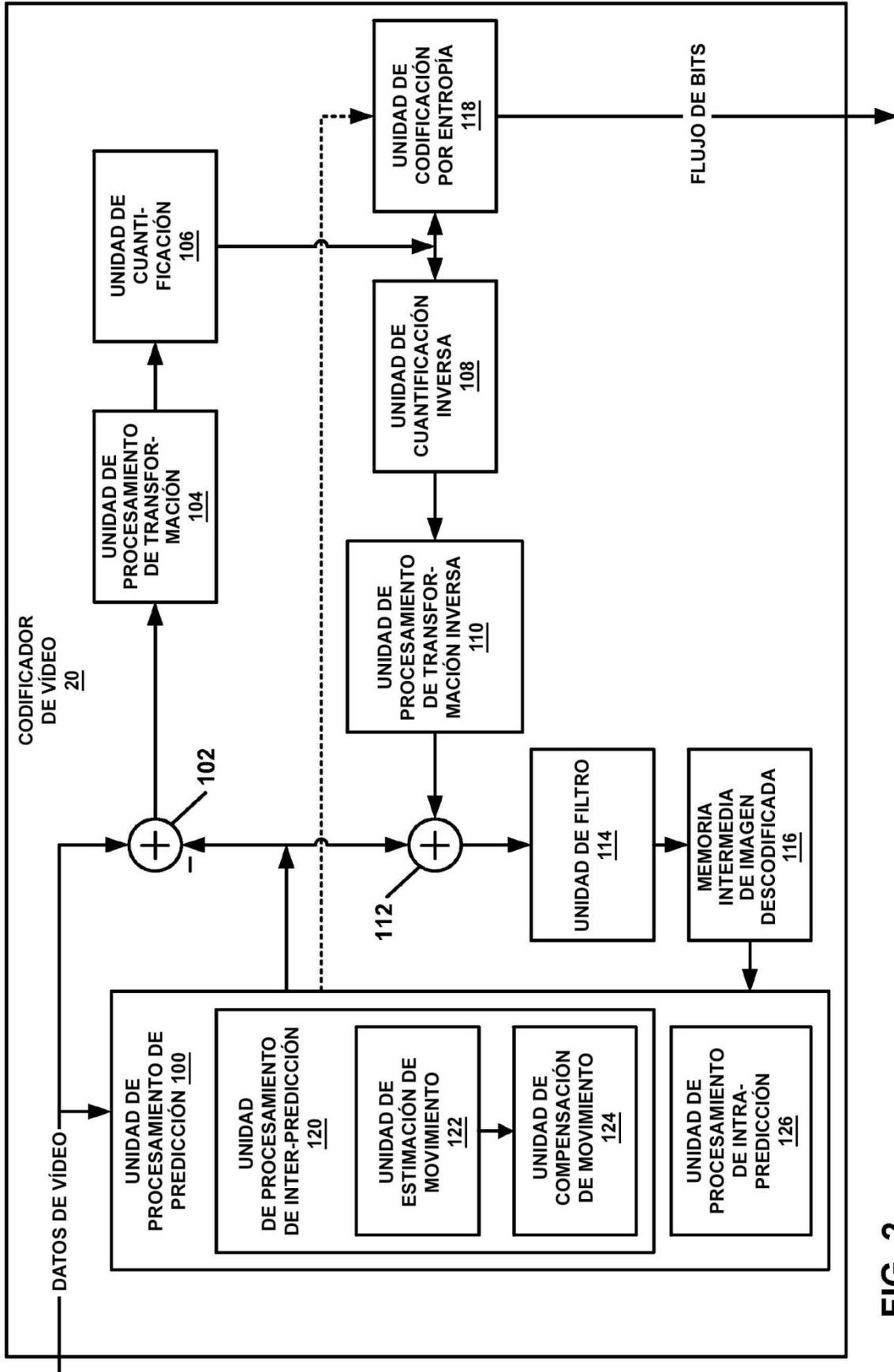


FIG. 2

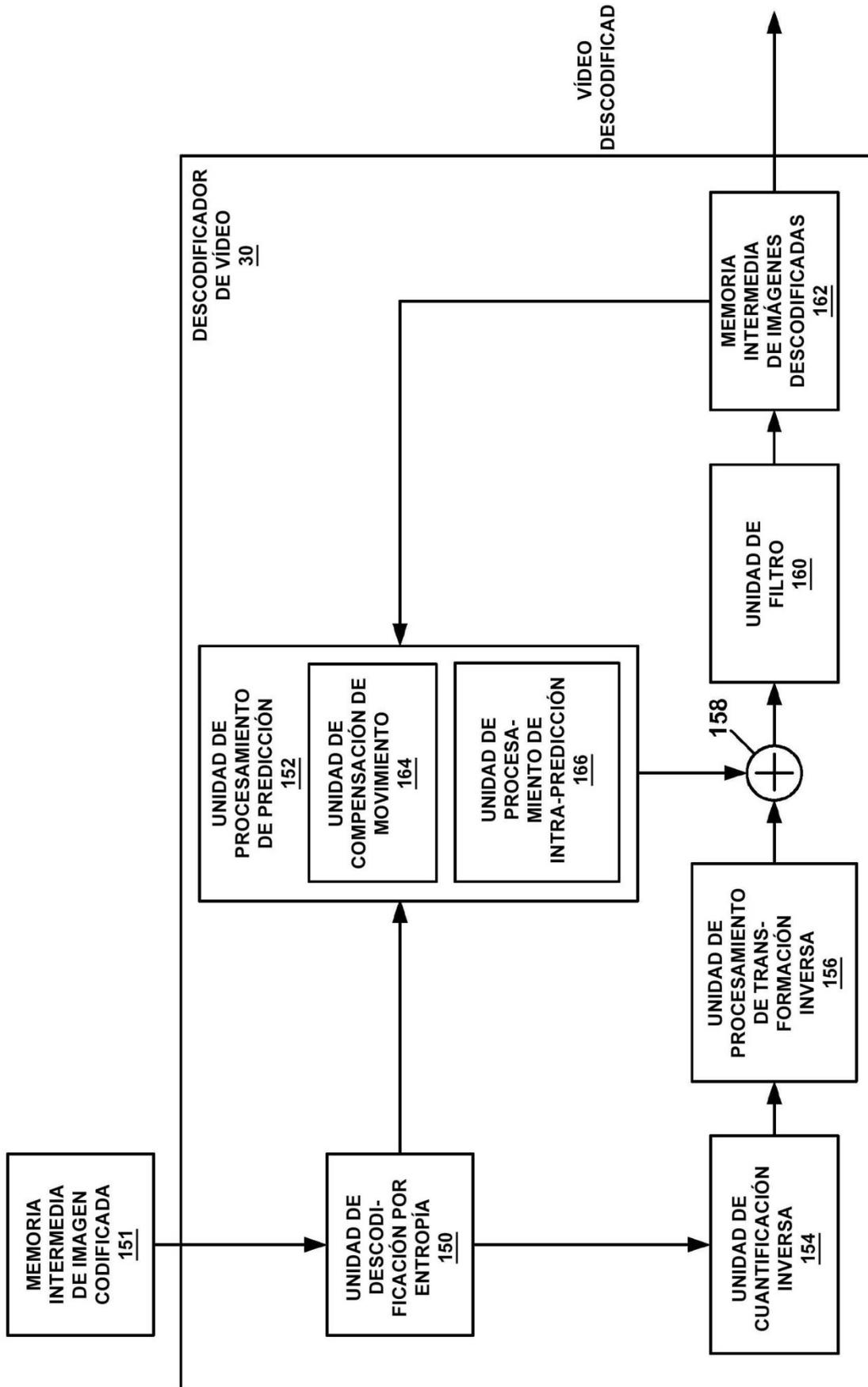


FIG. 3

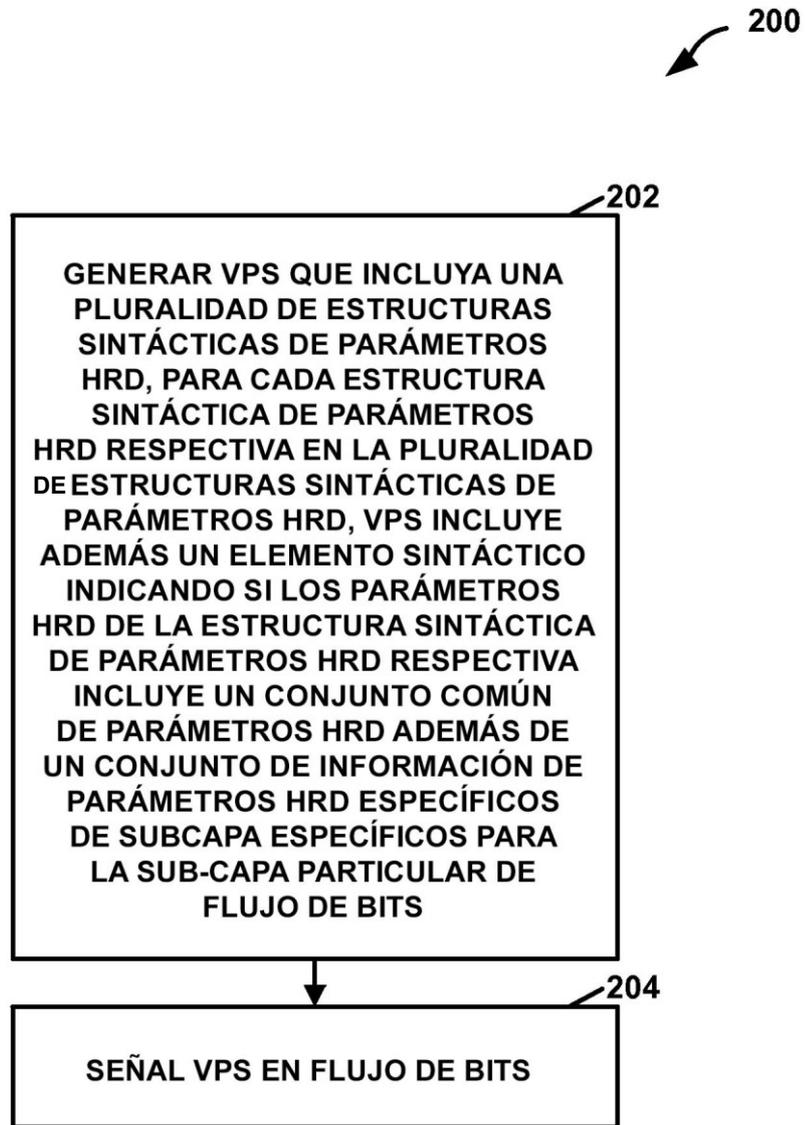


FIG. 4

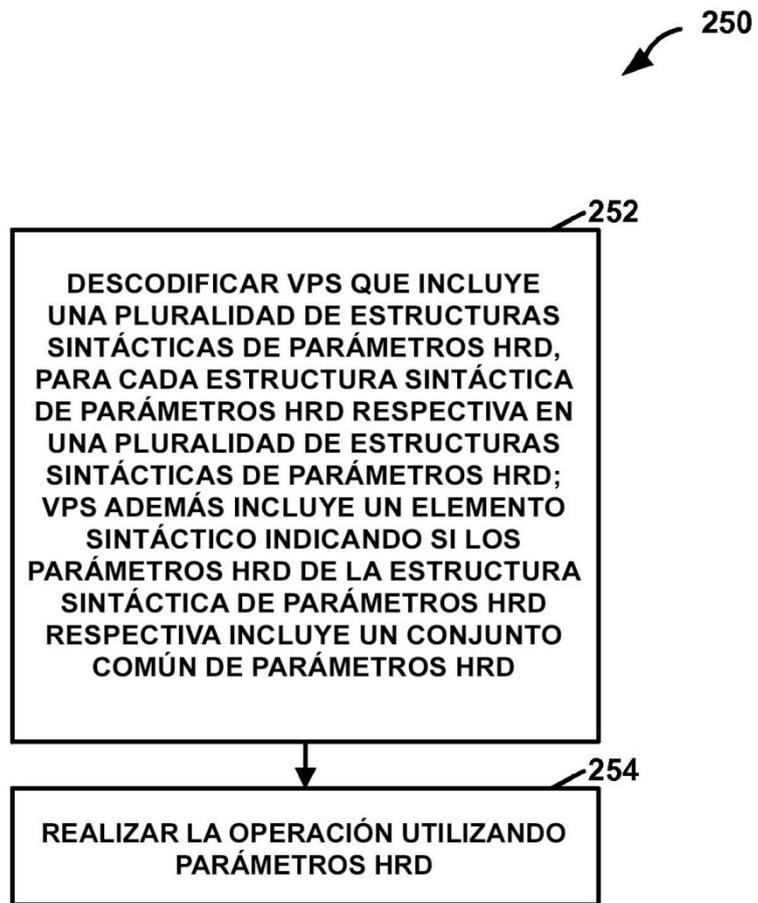


FIG. 5

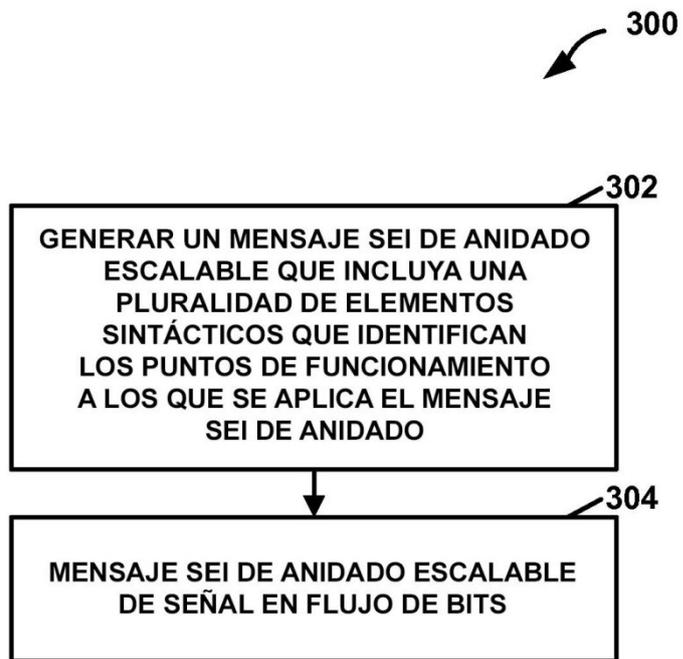


FIG. 6

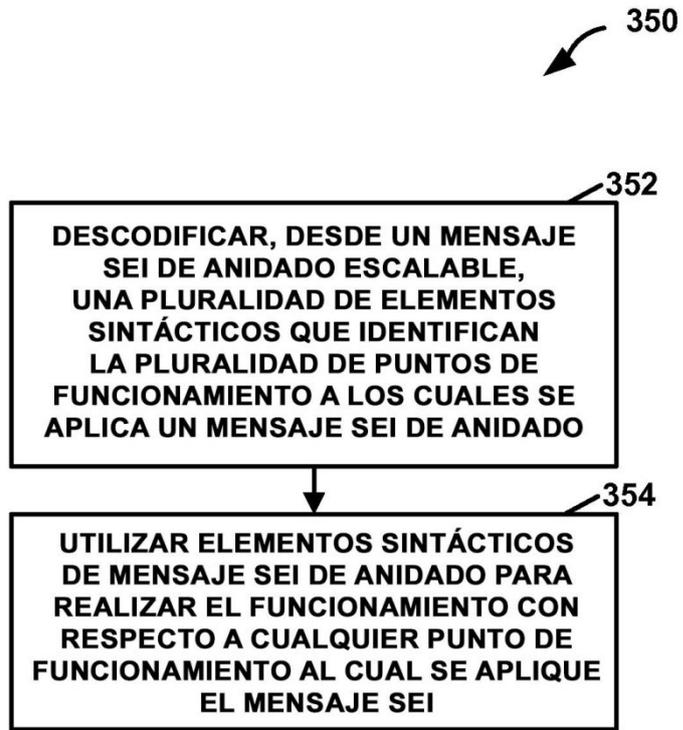


FIG. 7

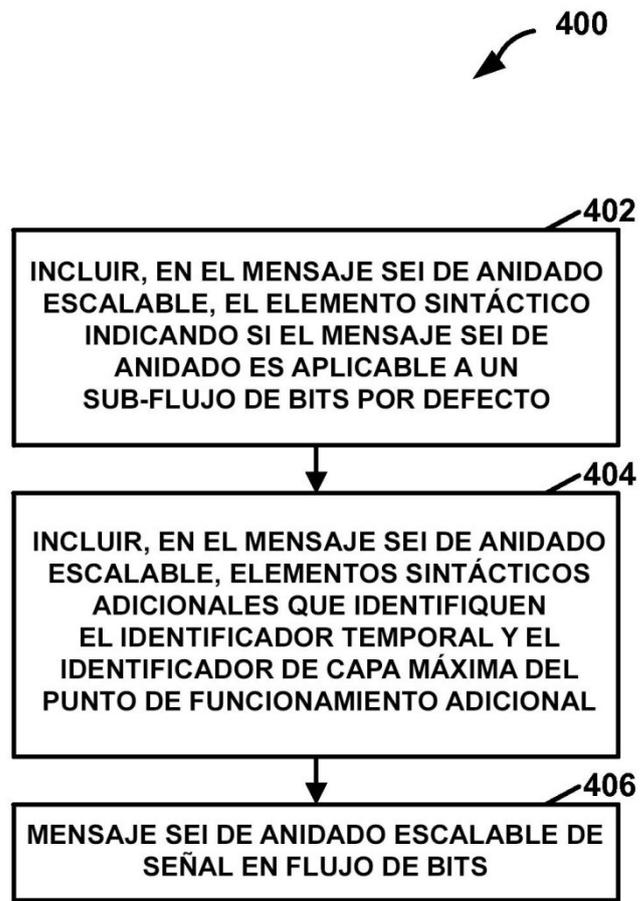


FIG. 8

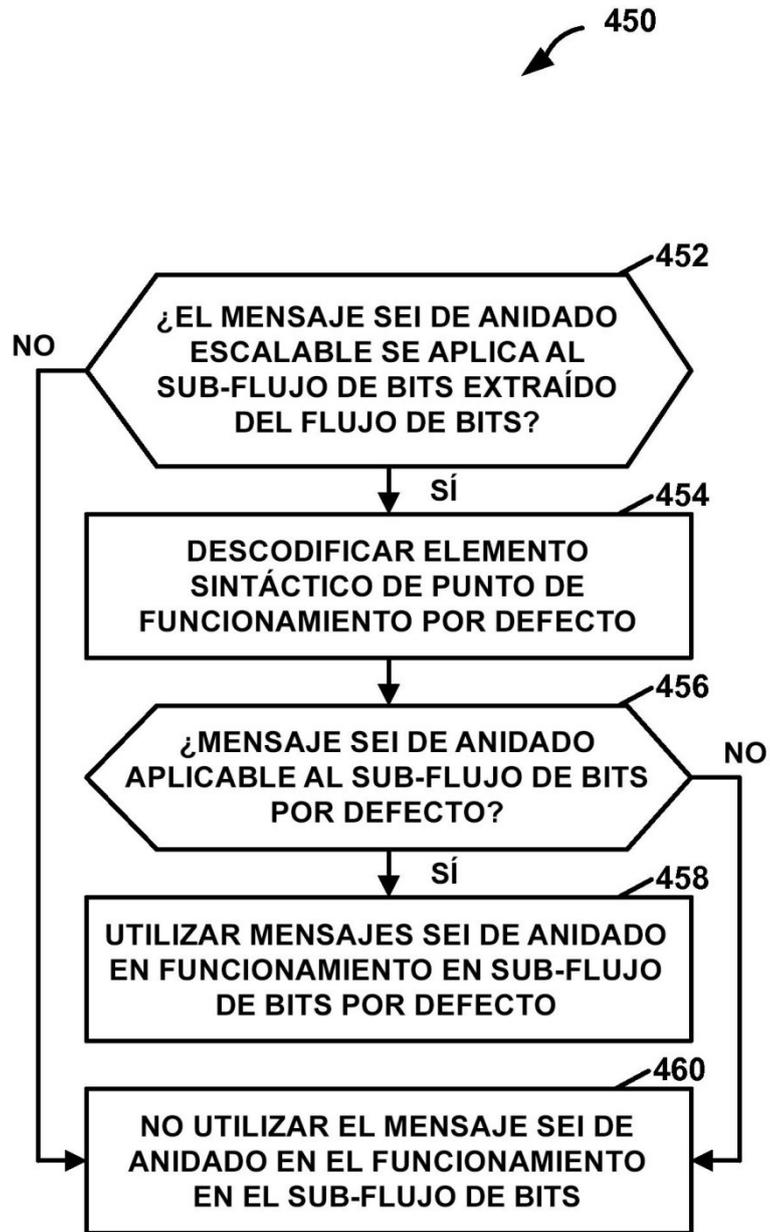


FIG. 9