

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 010 349**

51 Int. Cl.:

H04N 19/597 (2014.01)
H04N 19/117 (2014.01)
H04N 19/119 (2014.01)
H04N 19/172 (2014.01)
H04N 19/174 (2014.01)
H04N 19/70 (2014.01)
H04N 19/82 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **15.09.2020 PCT/US2020/050837**
 87 Fecha y número de publicación internacional: **01.04.2021 WO21061443**
 96 Fecha de presentación y número de la solicitud europea: **15.09.2020 E 20868321 (9)**
 97 Fecha y número de publicación de la concesión europea: **11.12.2024 EP 4022926**

54 Título: **Indicación de un segmento por subimagen en codificación de vídeo basada en subimágenes**

30 Prioridad:

23.09.2019 US 201962904481 P
24.09.2019 US 201962905122 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
02.04.2025

73 Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.00%)
Huawei Administration Building, Bantian,
Longgang District
Shenzhen, Guangdong 518129, CN

72 Inventor/es:

WANG, YE-KUI y
HENDRY, FNU

74 Agente/Representante:

PONS ARIÑO, Ángel

ES 3 010 349 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

Indicación de un segmento por subimagen en codificación de vídeo basada en subimágenes

5 Campo técnico

La presente descripción se refiere, en general, a codificación de vídeo, y está específicamente relacionada con la señalización de una indicación en codificación de vídeo basada en subimágenes.

10 Antecedentes

La cantidad de datos de vídeo necesarios para representar incluso un vídeo relativamente corto puede ser sustancial, lo que puede dar lugar a dificultades cuando los datos se van a transmitir en continuo o comunicar de otro modo a través de una red de comunicaciones con capacidad de ancho de banda limitada. Por lo tanto, los datos de vídeo generalmente se comprimen antes de comunicarse a través de las redes de telecomunicaciones modernas. El tamaño de un vídeo también podría ser un problema cuando el vídeo se almacena en un dispositivo de almacenamiento debido a que los recursos de memoria pueden ser limitados. Los dispositivos de compresión de vídeo a menudo usan software y/o hardware en la fuente para codificar los datos de vídeo antes de la transmisión o el almacenamiento, lo que reduce la cantidad de datos necesarios para representar imágenes de vídeo digital. Los datos comprimidos son recibidos a continuación en el destino por un dispositivo de descompresión de vídeo que decodifica los datos de vídeo. Con recursos de red limitados y demandas cada vez mayores de mayor calidad de vídeo, son deseables técnicas mejoradas de compresión y descompresión que mejoren la relación de compresión con poco o ningún sacrificio en la calidad de la imagen. El documento de HANNUKSELA (NOKIA) M ET AL: "AHG12: single_slice_per_subpic_flag", Joint Video Experts Team (JVET) de ITU-T SG 16 WP 3 y el documento de ISO/IEC JTC 1/SC 29/WG, no. JVET-P1024, 16 de octubre de 2019 (2019-10-01), Ginebra proponen single_slice_per_subpic_flag en el PPS.

30 Compendio

Un primer aspecto se refiere a un método implementado por un decodificador, que comprende recibir, por el decodificador, un flujo de bits que incluye un conjunto de parámetros de imagen (PPS) que incluye un indicador de segmento único por subimagen; determinar, por el decodificador, si el indicador de segmento único por subimagen tiene un primer valor o un segundo valor, donde el primer valor especifica que cada subimagen que se refiere al PPS incluye uno y solo un segmento rectangular, y donde el segundo valor especifica que cada subimagen que se refiere al PPS puede incluir uno o más segmentos rectangulares; y decodificar, por el decodificador, el uno y solo un segmento rectangular cuando el indicador de segmento único por subimagen tiene el primer valor y los uno o varios segmentos rectangulares cuando el único segmento por indicador de subimagen tiene el segundo valor para obtener una imagen decodificada. En lo que sigue, partes de la descripción y de los dibujos que se refieren a realizaciones anteriores que no necesariamente comprenden todas las características para implementar las realizaciones de la invención reivindicada no se representan como realizaciones de la invención, sino como ejemplos útiles para comprender las realizaciones de la invención.

El método da a conocer técnicas que utilizan un indicador en el flujo de bits para indicar si cada subimagen de una imagen contiene uno y sólo un segmento rectangular. Cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular, los elementos de sintaxis para señalar el número y la distribución de las subimágenes (por ejemplo, la posición y el tamaño de cada subimagen) se pueden omitir del flujo de bits porque las subimágenes son las mismas que los segmentos. Es decir, solo los elementos de sintaxis para los segmentos tienen que incluirse en el flujo de bits cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular. Esto reduce la cantidad de información transportada en el flujo de bits, lo que disminuye el uso de la memoria, del procesador y de los recursos de red y aumenta la eficiencia de la codificación. Por lo tanto, el codificador/decodificador (también conocido como "códec") en codificación de vídeo se mejora en relación con los códecs actuales. Como cuestión práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia de usuario cuando se envían, reciben y/o ven vídeos.

Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el indicador de segmento único por subimagen se designe como pps_single_slice_per_subpic_flag.

Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el primer valor es uno (1) y el segundo valor es cero (0).

Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el PPS comprende además un indicador del número de segmentos en imagen, y donde un valor del indicador

ES 3 010 349 T3

del número de segmentos en imagen más uno especifica un número de segmentos en cada imagen que se refiere al PPS.

5 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el indicador del número de segmentos en imagen se designa como `pps_num_slices_in_pic_minus1`.

Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que cada subimagen comprende una región rectangular de uno o más segmentos dentro de una imagen.

10 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer la generación de una imagen para visualización en base al uno y solo un segmento rectangular cuando el indicador de segmento único por subimagen tiene el primer valor y el uno o más segmentos rectangulares cuando el indicador de segmento único por subimagen tiene el segundo valor.

15 Un segundo aspecto se refiere a un método implementado por un codificador, comprendiendo el método ajustar, por el codificador, un indicador de segmento único por subimagen a un primer valor cuando cada subimagen que se refiere a un conjunto de parámetros de imagen (PPS) incluye uno y solo un segmento rectangular y a un segundo valor cuando cada subimagen que se refiere al PPS puede incluir uno o más segmentos rectangulares; codificar, por el codificador, el PPS que contiene el indicador de segmento único por subimagen en un flujo de bits; y almacenar, por el codificador, el flujo de bits para su comunicación hacia un decodificador.

20 El método da a conocer técnicas que utilizan un indicador en el flujo de bits para indicar si cada subimagen de una imagen contiene uno y sólo un segmento rectangular. Cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular, los elementos de sintaxis para señalar el número y la distribución de las subimágenes (por ejemplo, la posición y el tamaño de cada subimagen) se pueden omitir del flujo de bits porque las subimágenes son las mismas que los segmentos. Es decir, solo los elementos de sintaxis para los segmentos tienen que incluirse en el flujo de bits cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular. Esto reduce la cantidad de información transportada en el flujo de bits, lo que disminuye el uso de la memoria, del procesador y de los recursos de red y aumenta la eficiencia de la codificación. Por lo tanto, el codificador/decodificador (también conocido como "códec") en codificación de vídeo se mejora en relación con los códecs actuales. Como cuestión práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia de usuario cuando se envían, reciben y/o ven vídeos.

35 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el indicador de segmento único por subimagen se designe como `pps_single_slice_per_subpic_flag`.

40 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el primer valor es uno (1) y el segundo valor es cero (0).

Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el PPS comprende además un indicador del número de segmentos en imagen, y donde un valor del indicador del número de segmentos en imagen más uno especifica un número de segmentos en cada imagen que se refiere al PPS.

45 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el indicador del número de segmentos en imagen se designa como `pps_num_slices_in_pic_minus1`.

50 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que cada subimagen comprende una región rectangular de uno o más segmentos dentro de una imagen.

Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer transmitir el flujo de bits hacia un decodificador de vídeo.

55 Un tercer aspecto se refiere a un dispositivo de decodificación, que comprende un receptor configurado para recibir un flujo de bits que incluye un conjunto de parámetros de imagen (PPS) que contiene un indicador de segmento único por subimagen; una memoria acoplada al receptor, almacenando la memoria instrucciones; y un procesador acoplado a la memoria, el procesador configurado para ejecutar las instrucciones para hacer que el dispositivo de decodificación determine si el indicador de segmento único por subimagen tiene un primer valor o un segundo valor, donde el primer valor especifica que cada subimagen que se refiere al PPS incluye uno y solo un segmento rectangular, y donde el segundo valor especifica que cada subimagen que se refiere al PPS puede incluir uno o más segmentos rectangulares; y decodificar el uno y solo un segmento rectangular cuando el indicador de segmento único por subimagen tiene el primer valor y los uno o más segmentos rectangulares cuando el indicador de segmento único por subimagen tienen el segundo valor para obtener una imagen decodificada.

5 El dispositivo de decodificación da a conocer técnicas que utilizan un indicador en el flujo de bits para indicar si cada subimagen de una imagen contiene uno y sólo un segmento rectangular. Cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular, los elementos de sintaxis para señalar el número y la distribución de las subimágenes (por ejemplo, la posición y el tamaño de cada subimagen) se pueden omitir del flujo de bits porque las subimágenes son las mismas que los segmentos. Es decir, solo los elementos de sintaxis para los segmentos tienen que incluirse en el flujo de bits cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular. Esto reduce la cantidad de información transportada en el flujo de bits, lo que disminuye el uso de la memoria, del procesador y de los recursos de red y aumenta la eficiencia de la codificación. Por lo tanto, el codificador/decodificador (también conocido como "códec") en codificación de vídeo se mejora en relación con los códecs actuales. Como cuestión práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia de usuario cuando se envían, reciben y/o ven vídeos.

15 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el primer valor es uno (1) y el segundo valor es cero (0).

20 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el PPS comprende además un indicador del número de segmentos en imagen, y donde un valor del indicador del número de segmentos en imagen más uno especifica un número de segmentos en cada imagen que se refiere al PPS.

25 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el indicador del número de segmentos en imagen se designa como `pps_num_slices_in_pic_minus1`.

30 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el procesador está configurado además para generar una imagen para visualización en base al uno y solo un segmento rectangular cuando el indicador de segmento único por subimagen tiene el primer valor y el uno o más segmentos rectangulares cuando el indicador de segmento único por subimagen tiene el segundo valor.

35 Un cuarto aspecto se refiere a un dispositivo de codificación, que comprende una memoria que contiene instrucciones; un procesador acoplado a la memoria, el procesador configurado para implementar las instrucciones para hacer que el dispositivo de codificación codifique un flujo de bits que incluye un conjunto de parámetros de imagen (PPS) que contiene un indicador de segmento único por subimagen; ajustar el indicador de segmento único por subimagen a un primer valor cuando cada subimagen que se refiere al PPS incluye uno y solo un segmento rectangular y a un segundo valor cuando cada subimagen que se refiere al PPS puede incluir uno o más segmentos rectangulares; y un transmisor acoplado al procesador, el transmisor configurado para transmitir el flujo de bits hacia un decodificador de vídeo.

40 El dispositivo de codificación da a conocer técnicas que utilizan un indicador en el flujo de bits para indicar si cada subimagen de una imagen contiene uno y sólo un segmento rectangular. Cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular, los elementos de sintaxis para señalar el número y la distribución de las subimágenes (por ejemplo, la posición y el tamaño de cada subimagen) se pueden omitir del flujo de bits porque las subimágenes son las mismas que los segmentos. Es decir, solo los elementos de sintaxis para los segmentos tienen que incluirse en el flujo de bits cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular. Esto reduce la cantidad de información transportada en el flujo de bits, lo que disminuye el uso de la memoria, del procesador y de los recursos de red y aumenta la eficiencia de la codificación. Por lo tanto, el codificador/decodificador (también conocido como "códec") en codificación de vídeo se mejora en relación con los códecs actuales. Como cuestión práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia de usuario cuando se envían, reciben y/o ven vídeos.

55 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el primer valor es uno (1) y el segundo valor es cero (0).

60 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el PPS comprende además un indicador del número de segmentos en imagen, y donde un valor del indicador del número de segmentos en imagen más uno especifica un número de segmentos en cada imagen que se refiere al PPS.

Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que el indicador del número de segmentos en imagen se designa como `pps_num_slices_in_pic1`.

65 Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer que la memoria almacene el flujo de bits antes de transmitir el flujo de bits hacia el decodificador.

Un quinto aspecto se refiere a un aparato de codificación. El aparato de codificación incluye un receptor configurado para recibir una imagen a codificar o para recibir un flujo de bits a decodificar; un transmisor acoplado al receptor, el transmisor configurado para transmitir el flujo de bits a un decodificador o para transmitir una imagen decodificada a una pantalla; una memoria acoplada a al menos uno del receptor o del transmisor, la memoria configurada para almacenar instrucciones; y un procesador acoplado a la memoria, el procesador configurado para ejecutar las instrucciones almacenadas en la memoria para realizar cualquiera de los métodos dados a conocer en la presente memoria.

El aparato de codificación da a conocer técnicas que utilizan un indicador en el flujo de bits para indicar si cada subimagen de una imagen contiene uno y sólo un segmento rectangular. Cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular, los elementos de sintaxis para señalar el número y la distribución de las subimágenes (por ejemplo, la posición y el tamaño de cada subimagen) se pueden omitir del flujo de bits porque las subimágenes son las mismas que los segmentos. Es decir, solo los elementos de sintaxis para los segmentos tienen que incluirse en el flujo de bits cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular. Esto reduce la cantidad de información transportada en el flujo de bits, lo que disminuye el uso de la memoria, del procesador y de los recursos de red y aumenta la eficiencia de la codificación. Por lo tanto, el codificador/decodificador (también conocido como "códec") en codificación de vídeo se mejora en relación con los códecs actuales. Como cuestión práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia de usuario cuando se envían, reciben y/o ven vídeos.

Opcionalmente, en cualquiera de los aspectos anteriores, otra implementación del aspecto da a conocer una pantalla configurada para mostrar una imagen decodificada.

Un sexto aspecto se refiere a un sistema. El sistema incluye un codificador; y un decodificador en comunicación con el codificador, donde el codificador o el decodificador incluye el dispositivo de decodificación, el dispositivo de codificación o el aparato de codificación dado a conocer en la presente memoria.

El sistema da a conocer técnicas que utilizan un indicador en el flujo de bits para indicar si cada subimagen de una imagen contiene uno y sólo un segmento rectangular. Cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular, los elementos de sintaxis para señalar el número y la distribución de las subimágenes (por ejemplo, la posición y el tamaño de cada subimagen) se pueden omitir del flujo de bits porque las subimágenes son las mismas que los segmentos. Es decir, solo los elementos de sintaxis para los segmentos tienen que incluirse en el flujo de bits cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular. Esto reduce la cantidad de información transportada en el flujo de bits, lo que disminuye el uso de la memoria, del procesador y de los recursos de red y aumenta la eficiencia de la codificación. Por lo tanto, el codificador/decodificador (también conocido como "códec") en codificación de vídeo se mejora en relación con los códecs actuales. Como cuestión práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia de usuario cuando se envían, reciben y/o ven vídeos.

Un séptimo aspecto se refiere a un medio de codificación. El medio de codificación incluye medios de recepción configurados para recibir una imagen a codificar o para recibir un flujo de bits a decodificar; medios de transmisión acoplados a los medios de recepción, los medios de transmisión configurados para transmitir el flujo de bits a unos medios de decodificación o para transmitir una imagen decodificada a unos medios de pantalla; medios de almacenamiento acoplados a al menos uno de los medios de recepción o los medios de transmisión, estando configurados los medios de almacenamiento para almacenar instrucciones; y medios de procesamiento acoplados a los medios de almacenamiento, los medios de procesamiento configurados para ejecutar las instrucciones almacenadas en los medios de almacenamiento para realizar cualquiera de los métodos dados a conocer en la presente memoria.

El medio de codificación da a conocer técnicas que utilizan un indicador en el flujo de bits para indicar si cada subimagen de una imagen contiene uno y sólo un segmento rectangular. Cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular, los elementos de sintaxis para señalar el número y la distribución de las subimágenes (por ejemplo, la posición y el tamaño de cada subimagen) se pueden omitir del flujo de bits porque las subimágenes son las mismas que los segmentos. Es decir, solo los elementos de sintaxis para los segmentos tienen que incluirse en el flujo de bits cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular. Esto reduce la cantidad de información transportada en el flujo de bits, lo que disminuye el uso de la memoria, del procesador y de los recursos de red y aumenta la eficiencia de la codificación. Por lo tanto, el codificador/decodificador (también conocido como "códec") en codificación de vídeo se mejora en relación con los códecs actuales. Como cuestión práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia de usuario cuando se envían, reciben y/o ven vídeos.

Para mayor claridad, una cualquiera de las realizaciones anteriores puede combinarse con una o más de las otras realizaciones anteriores para crear una nueva realización dentro del alcance de la presente descripción.

5 Estas y otras características se entenderán más claramente a partir de la siguiente descripción detallada tomada junto con los dibujos adjuntos y las reivindicaciones.

Breve descripción de los dibujos

10 Para una comprensión más completa de esta descripción, se hace referencia a continuación a la siguiente descripción breve, tomada en relación con los dibujos adjuntos y la descripción detallada, donde los numerales de referencia iguales representan partes iguales.

La FIG. 1 es un diagrama de flujo de un método de ejemplo para codificar una señal de vídeo.

15 La FIG. 2 es un diagrama esquemático de un sistema de codificación y decodificación (códec) de ejemplo para codificación de vídeo.

La FIG. 3 es un diagrama esquemático que muestra un codificador de vídeo de ejemplo.

20 La FIG. 4 es un diagrama esquemático que muestra un decodificador de vídeo de ejemplo.

La FIG. 5 es un diagrama esquemático que ilustra un ejemplo de flujo de bits que contiene una secuencia de vídeo codificada.

25 La FIG. 6 es una realización de un método de decodificación de un flujo de bits de vídeo codificado.

La FIG. 7 es una realización de un método de codificación de un flujo de bits de vídeo codificado.

30 La FIG. 8 es un diagrama esquemático de un dispositivo de codificación de vídeo.

La FIG. 9 es un diagrama esquemático de una realización de un medio de codificación.

Descripción detallada

35 Debe entenderse desde el principio que, aunque a continuación se da a conocer una implementación ilustrativa de una o más realizaciones, los sistemas y/o métodos dados a conocer pueden implementarse usando cualquier número de técnicas, ya sean conocidas actualmente o existentes. La descripción no debe de ninguna manera limitarse a las implementaciones ilustrativas, dibujos, y técnicas ilustradas a continuación, que incluyen los diseños e implementaciones ejemplares ilustrados y descritos en la presente memoria, sino
40 que puede modificarse dentro del alcance de las reivindicaciones adjuntas junto con todos sus alcances de equivalencias.

45 Los siguientes términos se definen de la siguiente manera, a menos que se utilicen en un contexto contrario en la presente memoria. Específicamente, las siguientes definiciones están destinadas a proporcionar claridad adicional a la presente descripción. Sin embargo, los términos pueden describirse de manera diferente en diferentes contextos. Por consiguiente, las siguientes definiciones deben considerarse como un suplemento y no deben considerarse como una limitación de ninguna otra definición de las descripciones proporcionadas para dichos términos en la presente memoria.

50 Un flujo de bits es una secuencia de bits que incluye datos de vídeo que se comprimen para su transmisión entre un codificador y un decodificador. Un codificador es un dispositivo que se configura para emplear procesos de codificación para comprimir datos de vídeo en un flujo de bits. Un decodificador es un dispositivo que está configurado para emplear procesos de decodificación para reconstruir datos de vídeo de un flujo de bits para su visualización. Una imagen es un conjunto de muestras de luma y/o un conjunto de muestras de
55 croma que crean un cuadro o un campo del mismo. Una imagen que está siendo codificada o decodificada puede denominarse una imagen actual para mayor claridad de la explicación. Una imagen de referencia es una imagen que contiene muestras de referencia que se pueden usar al codificar otras imágenes por referencia de acuerdo con interpredicción y/o predicción entre capas. Una lista de imágenes de referencia es una lista de imágenes de referencia utilizadas para interpredicción y/o la predicción entre capas. Algunos
60 sistemas de codificación de vídeo utilizan dos listas de imágenes de referencia, que se pueden denotar como lista de imágenes de referencia uno y lista de imágenes de referencia cero. Una estructura de lista de imágenes de referencia es una estructura de sintaxis direccionable que contiene varias listas de imágenes de referencia. La interpredicción es un mecanismo de codificación de muestras de una imagen actual haciendo referencia a muestras indicadas en una imagen de referencia que es diferente de la imagen actual, donde la
65 imagen de referencia y la imagen actual están en la misma capa. Una entrada de estructura de lista de imágenes de referencia es una ubicación direccionable en una estructura de lista de imágenes de referencia

que indica una imagen de referencia asociada con una lista de imágenes de referencia. Una cabecera de segmento es una parte de un segmento codificado que contiene elementos de datos pertenecientes a todos los datos de vídeo dentro de una tesela representada en el segmento. Un segmento incluye cuatro lados que están conectados en ángulos rectos. Los cuatro lados incluyen dos pares de lados paralelos. Además, los
 5 lados en un par lateral paralelo son de igual longitud. Como tal, un segmento puede ser cualquier forma rectangular, donde un cuadrado es un caso especial de un rectángulo donde los cuatro lados tienen la misma longitud. Un segmento rectangular es un segmento que tiene dos pares de lados paralelos, donde los lados en cada lado paralelo son de igual longitud pero donde un par de los lados paralelos es más largo que el otro par de lados paralelos. Un conjunto de parámetros de secuencia (SPS) es un conjunto de parámetros que
 10 contiene datos relacionados con una secuencia de imágenes. Un conjunto de parámetros de imagen (PPS) es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a cero o más imágenes codificadas completas según lo determinado por un elemento de sintaxis que se encuentra en cada cabecera de imagen.

15 Un indicador es una variable o elemento de sintaxis de un solo bit que puede tomar uno de los dos valores posibles: 0 y 1. Una subimagen es una región rectangular de uno o más segmentos dentro de una imagen. Un identificador de subimagen (ID) es un número, letra u otro signo que identifica de forma única una subimagen. Los ID de subimagen (también conocidos como identificadores de tesela) se utilizan para
 20 identificar subimágenes particulares utilizando un índice de subimagen, que puede denominarse en la presente memoria como mapeo de ID de subimagen.

Una unidad de acceso (AU) es un conjunto de una o más imágenes codificadas asociadas con el mismo tiempo de visualización (por ejemplo, el mismo cómputo de orden de imagen) para entregar desde una memoria intermedia de imágenes decodificadas (DPB) (por ejemplo, para mostrar a un usuario). Un
 25 delimitador de unidad de acceso (AUD) es un indicador o estructura de datos utilizado para indicar el inicio de una AU o el límite entre AU. Una secuencia de vídeo decodificada es una secuencia de imágenes que han sido reconstruidas por un decodificador en preparación para mostrarse a un usuario.

Una secuencia de vídeo codificada (CVS) es una secuencia de unidades de acceso (AU) que incluye, en
 30 orden de decodificación, una AU de inicio de secuencia de vídeo codificada (CVSS), seguida de cero o más AU que no son AU CVSS, incluidas todas las AU posteriores, pero sin incluir ninguna AU posterior que sea una AU CVSS. Una AU CVSS es una AU en la que hay una unidad de predicción (PU) para cada capa especificada por el conjunto de parámetros de vídeo (VPS) y la imagen codificada en cada PU es una imagen de inicio de secuencia de vídeo de capa codificada (CLVSS). En una realización, cada imagen está dentro de
 35 una AU. Una PU es un conjunto de unidades de capa de abstracción de red (NAL) que están asociadas entre sí de acuerdo con una regla de clasificación especificada, son consecutivas en orden de decodificación y contienen exactamente una imagen codificada.

Se utilizan los siguientes acrónimos en la presente memoria: filtro de bucle adaptativo (ALF), bloque de árbol de codificación (CTB), unidad de árbol de codificación (CTU), unidad de codificación (CU), secuencia de vídeo codificada (CVS), equipo conjunto de expertos en vídeo (JVET), conjunto de teselas con restricción de movimiento (MCTS), unidad de transferencia máxima (MTU), capa de abstracción de red (NAL), cómputo de orden de imagen (POC), conjunto de parámetros de imagen (PPS), carga útil de secuencia de bytes sin procesar (RBSP), desplazamiento adaptativo de muestra (SAO), conjunto de parámetros de secuencia (SPS), predicción de vector de movimiento temporal (TMVP), codificación de vídeo versátil (VVC) y borrador de trabajo (WD).
 40
 45

Se pueden emplear muchas técnicas de compresión de vídeo para reducir el tamaño de los archivos de vídeo con una pérdida mínima de datos. Las técnicas de compresión de vídeo pueden incluir realizar predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia de datos en las secuencias de vídeo. Para codificación de vídeo basada en bloques, un segmento de vídeo (por ejemplo, una imagen de vídeo o una parte de una imagen de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, bloques de árbol de codificación (CTB), unidades de árbol de codificación (CTU), unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un segmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un segmento de predicción unidireccional (P) o predicción bidireccional (B) intercodificado de una imagen se pueden codificar utilizando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse cuadros y/o imágenes, y las imágenes de referencia pueden denominarse cuadros de referencia y/o imágenes de referencia. La predicción espacial o temporal tiene como resultado un bloque predictivo que representa un bloque de imagen. Los datos residuales representan diferencias de píxeles entre el bloque de imagen original y el bloque predictivo. Por consiguiente, un bloque intercodificado se codifica según un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica según un modo de intracodificación y los datos residuales. Para una
 50
 55
 60
 65

mayor compresión, los datos residuales pueden transformarse del dominio de píxeles a un dominio de transformada. Estos dan como resultado coeficientes de transformada residual, que pueden cuantificarse. Los coeficientes de transformada cuantificados pueden organizarse inicialmente en una matriz bidimensional. Los coeficientes de transformada cuantificados pueden ser escaneados para producir un vector unidimensional de coeficientes de transformada. Se puede aplicar codificación entrópica para conseguir aún más compresión. Estas técnicas de compresión de vídeo se discuten con mayor detalle a continuación.

Para garantizar que un vídeo codificado se pueda decodificar con precisión, el vídeo se codifica y decodifica de acuerdo con los estándares de codificación de vídeo correspondientes. Los estándares de codificación de vídeo incluyen H.261 del Sector de Estandarización de la Unión Internacional de Telecomunicaciones (ITU) (ITU-T), H.262 de ITU-T, parte 2 del Grupo de Expertos en Imágenes en Movimiento (MPEG)-1 de la Organización Internacional de Estandarización/Comisión Electrotécnica Internacional (ISO/IEC) o codificación de vídeo avanzada (AVC), Parte 2 de MPEG-2 ISO/IEC, H.263 de ITU-T, Parte 2 de MPEG-4 de ISO/IEC, también conocida como H.264 de ITU-T o Parte 10 de MPEG-4 de ISO/IEC, y codificación de vídeo de alta eficiencia (HEVC), también conocida como H.265 de ITU-T o Parte 2 de MPEG-H. La AVC incluye extensiones tales como codificación de vídeo escalable (SVC), codificación de vídeo multivista (MVC) y codificación de vídeo multivista más profundidad (MVC+D) y AVC de 3D (3D-AVC). HEVC incluye extensiones tales como HEVC escalable (SHVC), HEVC multivista (MV-HEVC) y HEVC 3D (3D-HEVC).

También hay un nuevo estándar de codificación de vídeo, denominado codificación de vídeo versátil (VVC), que está siendo desarrollado por el equipo conjunto de expertos en vídeo (JVET) de ITU-T e ISO/IEC. Si bien el estándar VVC tiene varios borradores de trabajo, un borrador de trabajo (WD) de VVC en particular, a saber, B. Bross, J. Chen y S. Liu, "Versatile Video Coding (Draft 5)", JVET-N1001-v3, 13.ª reunión de JVET, 27 de marzo de 2019 (borrador 5 de WC) se incorpora como referencia en su totalidad en la presente memoria.

El HEVC incluye cuatro esquemas de división de imágenes diferentes, a saber, segmentos regulares, segmentos dependientes, teselas y procesamiento paralelo de frente de onda (WPP), que se pueden aplicar para adaptación de tamaño de la unidad de transferencia máxima (MTU), el procesamiento paralelo y reducción del retraso de extremo a extremo.

Los segmentos regulares son similares a los de H.264/AVC. Cada segmento regular se encapsula en su propia unidad NAL, y se deshabilita la predicción en imagen (predicción intra-muestra, predicción de información de movimiento, predicción de modo de codificación) y la dependencia de codificación entrópica a través de los límites del segmento. Por lo tanto, un segmento regular puede reconstruirse independientemente de otros segmentos regulares dentro de la misma imagen (aunque aún puede haber interdependencias debido a operaciones de filtrado de bucle).

El segmento regular es la única herramienta que puede utilizarse para la paralelización que también está disponible, en forma prácticamente idéntica, en H.264/AVC. La paralelización basada en segmentos regulares no requiere mucha comunicación entre procesadores o entre núcleos (excepto para compartición de datos entre procesadores o entre núcleos para compensación de movimiento al decodificar una imagen codificada predictivamente, lo que generalmente es mucho más pesado que la compartición de datos entre procesadores o entre núcleos debido a la predicción en imagen). Sin embargo, por la misma razón, el uso de segmentos regulares puede incurrir en una sobrecarga de codificación sustancial debido al costo de bits de la cabecera de segmento y debido a la falta de predicción a través de los límites del segmento. Además, los segmentos regulares (en contraste con las otras herramientas mencionadas a continuación) también sirven como el mecanismo clave para que la división de flujo de bits coincida con los requisitos de tamaño de MTU, debido a la independencia en la imagen de los segmentos regulares y a que cada segmento regular está encapsulado en su propia unidad NAL. En muchos casos, el objetivo de la paralelización y el objetivo de adaptación de tamaño de la MTU contradicen las demandas de la distribución del segmento en una imagen. La comprensión de esta situación llevó al desarrollo de las herramientas de paralelización que se mencionan a continuación.

Los segmentos dependientes tienen cabeceras cortas de segmento y permiten dividir el flujo de bits en los límites del bloque de árbol sin romper ninguna predicción en la imagen. Básicamente, los segmentos dependientes proporcionan la fragmentación de los segmentos regulares en múltiples unidades NAL para proporcionar un retraso de extremo a extremo reducido al permitir que se envíe una parte de un segmento regular antes de que se termine la codificación de todo el segmento regular.

En WPP, la imagen se divide en filas individuales de bloques de árbol de codificación (CTB). Se permite que la decodificación entrópica y la predicción utilicen datos de CTB en otras divisiones. El procesamiento paralelo es posible mediante la decodificación paralela de las filas del CTB, donde el inicio de la decodificación de una fila del CTB se retrasa en dos CTB, a fin de garantizar que los datos relacionados con un CTB anterior y a la derecha del CTB correspondiente estén disponibles antes de que se decodifique el CTB correspondiente. Usando este inicio escalonado (que aparece como un frente de onda cuando se

representa gráficamente), la paralelización es posible con tantos procesadores/núcleos como filas CTB contenga la imagen. Debido a que se permite la predicción en imagen entre filas de bloques de árbol vecinas dentro de una imagen, la comunicación entre procesadores/núcleos requerida para permitir la predicción en imagen puede ser sustancial. La división WPP no tiene como resultado la producción de unidades NAL
 5 adicionales en comparación con cuando no se aplica, por lo que WPP no es una herramienta para la adaptación de tamaños MTU. Sin embargo, si se necesita adaptación de tamaño de MTU, se pueden usar segmentos regulares con WPP, con cierta sobrecarga de codificación.

Las teselas definen límites horizontales y verticales que dividen una imagen en columnas y filas de teselas. El
 10 orden de escaneo de los CTB se cambia para que sea local dentro de una tesela (en el orden de un barrido de trama de CTB de una tesela), antes de decodificar el CTB superior izquierdo de la siguiente tesela en el orden de barrido de trama de tesela de una imagen. Al igual que los segmentos regulares, las teselas rompen las dependencias de predicción en la imagen, así como las dependencias de decodificación entrópica. Sin embargo, no es necesario incluirlos en unidades NAL individuales (lo mismo que WPP en este sentido); por
 15 lo tanto, las teselas no se pueden usar para adaptación de tamaño de MTU. Cada tesela puede ser procesada por un procesador/núcleo, y la comunicación entre procesadores/entre núcleos requerida para la predicción en imagen entre unidades de procesamiento que decodifican teselas vecinas se limita a transportar la cabecera de segmento compartida en los casos en que un segmento abarca más de una tesela, y al filtrado de bucle relacionado con la compartición de muestras reconstruidas y metadatos. Cuando
 20 se incluye más de una tesela o sector WPP en un segmento, el desplazamiento de byte de punto de entrada para cada tesela o sector WPP que no sea el primero en el segmento se señala en la cabecera de segmento.

Por simplicidad, en el HEVC se han especificado restricciones a la aplicación de los cuatro diferentes esquemas de división de imágenes. Una secuencia de vídeo codificada dada no puede incluir tanto teselas
 25 como frentes de onda para la mayoría de los perfiles especificados en HEVC. Para cada segmento y tesela, tienen que cumplirse una o ambas de las siguientes condiciones: 1) todos los bloques de árbol codificados en un segmento pertenecen a la misma tesela; 2) todos los bloques de árbol codificados en una tesela pertenecen a al mismo segmento. Finalmente, un segmento de frente de onda contiene exactamente una fila CTB, y cuando WPP está en uso, si un segmento comienza dentro de una fila CTB, tiene que terminar en la
 30 misma fila CTB.

En el documento de salida JCT-VC, JCTVC-AC1005, J. Boyce, A. Ramasubramonian, R. Skupin, G. J. Sullivan, A. Tourapis, Y.-K. Wang (editores), "HEVC Additional Supplemental Enhancement Information (Draft
 35 4)", 24 de octubre de 2017, disponible públicamente en: http://phenix.int-evry.fr/jct/doc_end_user/documents/29_Macau/wg11/JCTVC-AC1005-v2.zip, se especifica una modificación reciente a HEVC. Con esta enmienda incluida, HEVC especifica tres mensajes de información de mejora suplementaria (SEI) relacionados con MCTS, a saber, el mensaje SEI temporal de MCTS, el mensaje SEI del conjunto de información de extracción de MCTS y el mensaje SEI de anidamiento de información de extracción de MCTS.

El mensaje SEI temporal de MCTS indica la existencia de MCTS en el flujo de bits y señala los MCTS. Para cada MCTS, los vectores de movimiento están restringidos para apuntar a ubicaciones de muestra completa dentro del MCTS y a ubicaciones de muestra fraccionada que requieren solo ubicaciones de muestra completa dentro del MCTS para la interpolación, y no se permite el uso de candidatos de vectores de
 45 movimiento para la predicción de vectores de movimiento temporal derivados de bloques fuera del MCTS. De esta manera, cada MCTS puede ser decodificado independientemente sin la existencia de teselas no incluidas en el MCTS.

El mensaje SEI de conjuntos de información de extracción de MCTS proporciona información suplementaria que se puede utilizar en la extracción de subflujos de bits de MCTS (especificada como parte de la semántica del mensaje SEI) para generar un flujo de bits conforme para un conjunto MCTS. La información consiste en una serie de conjuntos de información de extracción, cada uno definiendo un número de conjuntos MCTS y conteniendo bytes RBSP de los conjuntos de parámetros de vídeo de reemplazo (VPS), SPS y PPS que se utilizarán durante el proceso de extracción de subflujo de bits MCTS. Cuando se extrae un subflujo de bits de
 50 acuerdo con el proceso de extracción de subflujo de bits MCTS, los conjuntos de parámetros (VPS, SPS y PPS) tienen que reescribirse o reemplazarse, las cabeceras de segmento tienen que actualizarse ligeramente porque uno o todos los elementos de sintaxis relacionados con la dirección de segmento (incluidos `first_slice_segment_in_pic_flag` y `slice_segment_address`) generalmente necesitarían tener diferentes valores.

En la última especificación de borrador de VVC, una imagen puede dividirse en múltiples subimágenes, cada una cubriendo una región rectangular y conteniendo un número entero de segmentos completos. La división de subimágenes persiste en todas las imágenes dentro de una secuencia de vídeo codificada (CVS), y la información de división (es decir, las posiciones y tamaños de la subimagen) se señala en el SPS. Una subimagen puede indicarse como codificada sin usar valores de muestra de ninguna otra subimagen para la
 60 compensación de movimiento.

En la contribución JVET JVET-00141, que está disponible públicamente en: http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/13_Marrakech/wg11/JVET-M0261-v1.zip, el diseño de subimagen es similar al de la última especificación de borrador de VVC con algunas diferencias, una de las cuales es la señalización explícita de un ID de subimagen para cada subimagen en el SPS, así como la señalización del ID de subimagen en cada cabecera de segmento para permitir la extracción de subflujo de bits basada en subimagen sin la necesidad de cambiar las unidades NAL de segmento codificadas. En este enfoque, el ID de subimagen para cada subimagen se mantiene sin cambios en todas las imágenes de una CVS.

En un caso de uso común, cada subimagen contiene solo un segmento. Sin embargo, falta la capacidad de indicar el caso de uso común, lo que es indeseable. De hecho, saber que cada subimagen contiene solo un segmento podría resultar útil. Por ejemplo, cuando se conoce la distribución de los segmentos en las imágenes, los bits para señalar la distribución de la subimagen podrían omitirse por completo del flujo de bits, ya que la distribución de las subimágenes sería exactamente la misma que la distribución de los segmentos.

En el presente documento se dan a conocer técnicas que utilizan un indicador en el flujo de bits para indicar si cada subimagen de una imagen contiene uno y sólo un segmento rectangular. Cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular, los elementos de sintaxis para señalar el número y la distribución de las subimágenes (por ejemplo, la posición y el tamaño de cada subimagen) se pueden omitir del flujo de bits porque las subimágenes son las mismas que los segmentos. Es decir, solo los elementos de sintaxis para los segmentos tienen que incluirse en el flujo de bits cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular. Esto reduce la cantidad de información transportada en el flujo de bits, lo que disminuye el uso de la memoria, del procesador y de los recursos de red y aumenta la eficiencia de la codificación. Por lo tanto, el codificador/decodificador (también conocido como "códec") en codificación de vídeo se mejora en relación con los códecs actuales. Como cuestión práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia de usuario cuando se envían, reciben y/o ven vídeos.

La FIG. 1 es un diagrama de flujo de un método 100 de ejemplo para codificar una señal de vídeo. Específicamente, una señal de vídeo se codifica en un codificador. El proceso de codificación comprime la señal de vídeo empleando varios mecanismos para reducir el tamaño del archivo de vídeo. Un tamaño de archivo más pequeño permite que el archivo de vídeo comprimido se transmita hacia un usuario, al tiempo que reduce la sobrecarga de ancho de banda asociada. El decodificador luego decodifica el archivo de vídeo comprimido para reconstruir la señal de vídeo original para mostrarla a un usuario final. El proceso de decodificación generalmente refleja el proceso de codificación para permitir que el decodificador reconstruya consistentemente la señal de vídeo.

En la etapa 101, la señal de vídeo se introduce en el codificador. Por ejemplo, la señal de vídeo puede ser un archivo de vídeo sin comprimir almacenado en la memoria. Como otro ejemplo, el archivo de vídeo puede ser capturado por un dispositivo de captura de vídeo, tal como una cámara de vídeo, y codificado para soportar la transmisión en continuo del vídeo. El archivo de vídeo puede incluir tanto un componente de audio como un componente de vídeo. El componente de vídeo contiene una serie de cuadros de imagen que, cuando se ven en una secuencia, dan la impresión visual de movimiento. Los cuadros contienen píxeles que se expresan en términos de luz, denominados en la presente memoria componentes de luma (o muestras de luma), y color, que se denominan componentes de croma (o muestras de color). En algunos ejemplos, los cuadros también pueden contener valores de profundidad para soportar la visualización tridimensional.

En la etapa 103, el vídeo se divide en bloques. La división incluye subdividir los píxeles en cada cuadro en bloques cuadrados y/o rectangulares para su compresión. Por ejemplo, en codificación de vídeo de alta eficiencia (HEVC) (también conocido como H.265 y MPEG-H Parte 2), el cuadro se puede dividir primero en unidades de árbol de codificación (CTU), que son bloques de un tamaño predefinido (por ejemplo, sesenta y cuatro píxeles por sesenta y cuatro píxeles). Las CTU contienen tanto muestras de luma como de croma. Se pueden emplear árboles de codificación para dividir las CTU en bloques y luego subdividir recursivamente los bloques hasta que se consigan configuraciones que soporten codificación adicional. Por ejemplo, los componentes de luma de un cuadro se pueden subdividir hasta que los bloques individuales contengan valores de iluminación relativamente homogéneos. Además, los componentes de croma de un cuadro se pueden subdividir hasta que los bloques individuales contengan valores de color relativamente homogéneos. Por consiguiente, los mecanismos de división varían dependiendo del contenido de los cuadros de vídeo.

En la etapa 105, se emplean varios mecanismos de compresión para comprimir los bloques de imagen divididos en la etapa 103. Por ejemplo, se puede emplear interpredicción y/o intrapredicción. La interpredicción está diseñada para aprovechar el hecho de que los objetos de una escena común tienden a aparecer en cuadros sucesivos. Por consiguiente, un bloque que representa un objeto en un cuadro de referencia no necesita describirse repetidamente en cuadros posteriores. Específicamente, un objeto, tal como una mesa, puede permanecer en una posición constante sobre múltiples cuadros. Por lo tanto, la mesa

se describe una vez y los cuadros posteriores pueden referirse al cuadro de referencia. Se pueden emplear mecanismos de coincidencia de patrones para poner en correspondencia objetos sobre múltiples cuadros. Además, los objetos en movimiento pueden representarse en múltiples cuadros, por ejemplo, debido al movimiento del objeto o al movimiento de la cámara. Como un ejemplo particular, un vídeo puede mostrar un automóvil que se mueve a través de la pantalla sobre múltiples cuadros. Se pueden emplear vectores de movimiento para describir dicho movimiento. Un vector de movimiento es un vector bidimensional que proporciona un desplazamiento desde las coordenadas de un objeto en un cuadro hasta las coordenadas del objeto en un cuadro de referencia. Como tal, la interpredicción puede codificar un bloque de imagen en un cuadro actual como un conjunto de vectores de movimiento que indican un desplazamiento de un bloque correspondiente en un cuadro de referencia.

La intrapredicción codifica bloques en un cuadro común. La intrapredicción aprovecha el hecho de que los componentes de luma y de croma tienden a agruparse en un cuadro. Por ejemplo, una mancha de verde en una parte de un árbol tiende a estar situada junto a manchas similares de verde. La intrapredicción emplea múltiples modos de predicción direccional (por ejemplo, treinta y tres en HEVC), un modo plano y un modo de corriente continua (CC). Los modos direccionales indican que un bloque actual es similar/el mismo que las muestras de un bloque vecino en una dirección correspondiente. El modo plano indica que una serie de bloques a lo largo de una fila/columna (por ejemplo, un plano) se pueden interpolar en base a bloques vecinos en los bordes de la fila. El modo plano, en efecto, indica una transición suave de luz/color a través de una fila/columna empleando una pendiente relativamente constante en valores cambiantes. El modo DC se emplea para el suavizado de límites e indica que un bloque es similar/el mismo que un valor promedio asociado con muestras de todos los bloques vecinos asociados con las direcciones angulares de los modos de predicción direccional. Por consiguiente, los bloques de intrapredicción pueden representar bloques de imagen como varios valores de modo de predicción relacional en lugar de los valores reales. Además, los bloques de interpredicción pueden representar bloques de imagen como valores de vector de movimiento en lugar de los valores reales. En cualquier caso, los bloques de predicción pueden no representar exactamente los bloques de imagen en algunos casos. Cualquier diferencia se almacena en bloques residuales. Pueden aplicarse transformadas a los bloques residuales para comprimir más el archivo.

En la etapa 107, pueden aplicarse varias técnicas de filtrado. En HEVC, los filtros se aplican según un esquema de filtrado en bucle. La predicción basada en bloques analizada anteriormente puede tener como resultado la creación de imágenes en bloque en el decodificador. Además, el esquema de predicción basado en bloques puede codificar un bloque y luego reconstruir el bloque codificado para su uso posterior como un bloque de referencia. El esquema de filtrado en bucle aplica iterativamente filtros de supresión de ruido, filtros de desbloqueo, filtros de bucle adaptativo y filtros de desplazamiento adaptativo de muestra (SAO) a los bloques/cuadros. Estos filtros mitigan dichos artefactos de bloque para que el archivo codificado pueda reconstruirse con precisión. Además, estos filtros mitigan los artefactos en los bloques de referencia reconstruidos, para que sea menos probable que los artefactos creen artefactos adicionales en los bloques posteriores que se codifican en base a los bloques de referencia reconstruidos.

Una vez que la señal de vídeo ha sido dividida, comprimida y filtrada, los datos resultantes se codifican en un flujo de bits en la etapa 109. El flujo de bits incluye los datos analizados anteriormente, así como cualesquiera datos de señalización deseados para soportar la reconstrucción adecuada de la señal de vídeo en el decodificador. Por ejemplo, dichos datos pueden incluir datos de división, datos de predicción, bloques residuales y varios indicadores que proporcionan instrucciones de codificación al decodificador. El flujo de bits puede ser almacenado en memoria para su transmisión hacia un decodificador bajo demanda. El flujo de bits también puede ser difundido y/o multidifundido hacia una pluralidad de decodificadores. La creación del flujo de bits es un proceso iterativo. Por consiguiente, las etapas 101, 103, 105, 107 y 109 pueden ocurrir de manera continua y/o simultánea sobre muchos cuadros y bloques. El orden que se muestra en la FIG. 1 se presenta para mayor claridad y facilidad del análisis, y no tiene la intención de limitar el proceso de codificación de vídeo a un orden particular.

El decodificador recibe el flujo de bits y comienza el proceso de decodificación en la etapa 111. Específicamente, el decodificador emplea un esquema de decodificación entrópica para convertir el flujo de bits en correspondientes datos de sintaxis y de vídeo. El decodificador emplea los datos de sintaxis del flujo de bits para determinar las divisiones para los cuadros en la etapa 111. La división debe coincidir con los resultados de la división de bloques en la etapa 103. A continuación se describe la codificación/decodificación entrópica como se emplea en la etapa 111. El codificador adopta muchas elecciones durante el proceso de compresión, tales como seleccionar esquemas de división de bloques entre varias opciones posibles basadas en el posicionamiento espacial de los valores en la o las imágenes de entrada. La señalización de las opciones exactas puede emplear un gran número de tramos. Como se usa en la presente memoria, un tramo es un valor binario que se trata como variable (por ejemplo, un valor de bits que puede variar dependiendo del contexto). La codificación entrópica permite al codificador desechar cualquier opción que claramente no sea viable para un caso particular, dejando un conjunto de opciones permitidas. A cada opción permitida se le asigna una palabra de código. La longitud de las palabras de código se basa en el número de opciones permitidas (por ejemplo, un tramo para dos opciones, dos tramos para tres a cuatro opciones, etc.).

A continuación, el codificador codifica la palabra de código para la opción seleccionada. Este esquema reduce el tamaño de las palabras de código, ya que las palabras de código son tan grandes como se desee para indicar de forma única una selección de un pequeño subconjunto de opciones permisibles en lugar de indicar de forma única la selección desde un conjunto potencialmente grande de todas las opciones posibles.

5 El decodificador luego decodifica la selección determinando el conjunto de opciones permitidas de una manera similar al codificador. Determinando el conjunto de opciones permitidas, el decodificador puede leer la palabra de código y determinar la selección realizada por el codificador.

10 En la etapa 113, el decodificador realiza la decodificación de bloques. Específicamente, el decodificador emplea transformadas inversas para generar bloques residuales. Luego, el decodificador emplea los bloques residuales y los bloques de predicción correspondientes para reconstruir los bloques de imagen según la división. Los bloques de predicción pueden incluir tanto bloques de intrapredicción como bloques de interpredicción tal como se generan en el codificador en la etapa 105. Los bloques de imagen reconstruidos se colocan entonces en cuadros de una señal de vídeo reconstruida según los datos de división

15 determinados en la etapa 111. La sintaxis para la etapa 113 también se puede señalar en el flujo de bits mediante codificación entrópica como se analizó anteriormente.

En la etapa 115, el filtrado se realiza en los cuadros de la señal de vídeo reconstruida de una manera similar a la etapa 107 en el codificador. Por ejemplo, se pueden aplicar filtros de supresión de ruido, filtros de desbloqueo, filtros de bucle adaptativo y filtros SAO a los cuadros para eliminar artefactos de bloqueo. Una vez que se filtran los cuadros, la señal de vídeo se puede entregar a una pantalla en la etapa 117 para que la vea un usuario final.

20

La FIG. 2 es un diagrama esquemático de un sistema 200 de codificación y decodificación (códec) de ejemplo para codificación de vídeo. Específicamente, el sistema de códec 200 proporciona funcionalidad para soportar la implementación del método 100. El sistema de códec 200 se generaliza para representar componentes empleados tanto en un codificador como en un decodificador. El sistema de códec 200 recibe y divide una señal de vídeo como se analiza con respecto a las etapas 101 y 103 en el método 100, lo que tiene como resultado una señal de vídeo dividida 201. El sistema de códec 200 comprime a continuación la señal de vídeo dividida 201 en un flujo de bits codificado cuando actúa como un codificador como se analizó con respecto a las etapas 105, 107 y 109 en el método 100. Cuando actúa como un decodificador, el sistema de códec 200 genera una señal de vídeo de salida a partir del flujo de bits, como se analiza con respecto a las etapas 111, 113, 115 y 117 en el método 100. El sistema de códec 200 incluye un componente 211 de control de codificador general, un componente 213 de escalado de transformada y cuantificación, un componente 215 de estimación intraimagen, un componente 217 de predicción intraimagen, un componente 219 de compensación de movimiento, un componente 221 de estimación de movimiento, un componente 229 de escalado y transformada inversa, un componente 227 de análisis de control de filtros, un componente 225 de filtros en bucle, un componente 223 de memoria intermedia de imágenes decodificadas, y un componente 231 de formateo de cabecera y de codificación aritmética binaria adaptativa de contexto (CABAC). Dichos componentes se acoplan como se muestra. En la FIG. 2, las líneas negras indican el movimiento de los datos a codificar/decodificar, mientras que las líneas discontinuas indican el movimiento de los datos de control que controlan el funcionamiento de otros componentes. Todos los componentes del sistema de códec 200 pueden estar presentes en el codificador. El decodificador puede incluir un subconjunto de los componentes del sistema de códec 200. Por ejemplo, el decodificador puede incluir el componente 217 de predicción intraimagen, el componente 219 de compensación de movimiento, el componente 229 de escalado y transformada inversa, el componente 225 de filtros en bucle y el componente 223 de memoria intermedia de imágenes decodificadas. Estos componentes se describen a continuación.

25

30

35

40

45

La señal de vídeo dividida 201 es una secuencia de vídeo capturada que ha sido dividida en bloques de píxeles por un árbol de codificación. Un árbol de codificación emplea varios modos de división para subdividir un bloque de píxeles en bloques más pequeños de píxeles. Estos bloques se pueden subdividir más, en bloques más pequeños. Los bloques pueden denominarse nodos en el árbol de codificación. Los nodos padres mayores se dividen en nodos hijos más pequeños. El número de veces que se subdivide un nodo se denomina profundidad del árbol de codificación/nodo. Los bloques divididos pueden estar incluidos en unidades de codificación (CU) en algunos casos. Por ejemplo, una CU puede ser una subporción de una CTU que contiene un bloque de luma, uno o varios bloques de croma de diferencia roja (Cr) y uno o varios bloques de croma de diferencia azul (Cb) junto con las instrucciones de sintaxis correspondientes para la CU. Los modos de división pueden incluir un árbol binario (BT), un árbol triple (TT) y un árbol cuádruple (QT) empleados para dividir un nodo en dos, tres o cuatro nodos hijos, respectivamente, de formas variables dependiendo de los modos de división empleados. La señal de vídeo dividida 201 se reenvía al componente 211 de control de codificador general, al componente 213 de escalado de transformada y cuantificación, al componente 215 de estimación intraimagen, al componente 227 de análisis de control de filtros y al componente 221 de estimación de movimiento para su compresión.

50

55

60

65 El componente 211 de control de codificador general está configurado para tomar decisiones relacionadas con la codificación de las imágenes de la secuencia de vídeo en el flujo de bits según las restricciones de la

aplicación. Por ejemplo, el componente 211 de control de codificador general gestiona la optimización de la tasa de bits/tamaño de flujo de bits frente a la calidad de reconstrucción. Dichas decisiones pueden tomarse basadas en la disponibilidad de espacio de almacenamiento/ancho de banda y las solicitudes de resolución de imágenes. El componente 211 de control de codificador general también gestiona la utilización de memoria intermedia a la luz de la velocidad de transmisión para mitigar los problemas de subdesbordamiento y desbordamiento de la memoria intermedia. Para gestionar estos problemas, el componente 211 de control de codificador general gestiona la división, la predicción y el filtrado por los otros componentes. Por ejemplo, el componente 211 de control de codificador general puede aumentar dinámicamente la complejidad de compresión para aumentar la resolución y aumentar el uso del ancho de banda o disminuir la complejidad de compresión para disminuir la resolución y el uso del ancho de banda. Por lo tanto, el componente 211 de control de codificador general controla los otros componentes del sistema de códec 200 para equilibrar la calidad de reconstrucción de la señal de vídeo en relación con la tasa de bits. El componente 211 de control de codificador general crea datos de control, que controlan el funcionamiento de los otros componentes. Los datos de control también se reenvían al componente 231 de formateo de cabecera y CABAC para codificarse en el flujo de bits para señalar parámetros de decodificación en el decodificador.

La señal de vídeo dividida 201 también se envía al componente 221 de estimación de movimiento y al componente 219 de compensación de movimiento para interpredicción. El cuadro o segmento de la señal de vídeo dividida 201 se puede dividir en múltiples bloques de vídeo. El componente 221 de estimación de movimiento y el componente 219 de compensación de movimiento realizan codificación interpredictiva del bloque de vídeo recibido con respecto a uno o más bloques en uno o más cuadros de referencia para proporcionar predicción temporal. El sistema de códec 200 puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.

El componente 221 de estimación de movimiento y el componente 219 de compensación de movimiento pueden estar altamente integrados, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por el componente 221 de estimación de movimiento, es el proceso de generar vectores de movimiento, que estiman el movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un objeto codificado en relación con un bloque predictivo. Un bloque predictivo es un bloque que se encuentra que coincide estrechamente con el bloque a codificar, en términos de diferencia de píxeles. Un bloque predictivo también puede denominarse bloque de referencia. Dicha diferencia de píxeles puede determinarse mediante la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD) u otras métricas de diferencia. HEVC emplea varios objetos codificados, incluyendo una CTU, bloques de árbol de codificación (CTB) y CU. Por ejemplo, una CTU puede dividirse en CTB, que luego pueden dividirse en BC para su inclusión en las CU. Una CU se puede codificar como una unidad de predicción (PU) que contiene datos de predicción y/o una unidad de transformada (TU) que contiene datos residuales transformados para la CU. El componente 221 de estimación de movimiento genera vectores de movimiento, PU y TU utilizando un análisis de tasa-distorsión como parte de un proceso de optimización de tasa-distorsión. Por ejemplo, el componente 221 de estimación de movimiento puede determinar múltiples bloques de referencia, múltiples vectores de movimiento, etc. para un bloque/cuadro actual, y puede seleccionar los bloques de referencia, vectores de movimiento, etc. que tengan las mejores características de tasa-distorsión. Las mejores características de tasa-distorsión equilibran la calidad de la reconstrucción de vídeo (por ejemplo, la cantidad de pérdida de datos por compresión) con la eficiencia de la codificación (por ejemplo, el tamaño de la codificación final).

En algunos ejemplos, el sistema de códec 200 puede calcular valores para posiciones de píxeles de subenteros de imágenes de referencia almacenadas en el componente 223 de memoria intermedia de imágenes decodificadas. Por ejemplo, el sistema de códec 200 de vídeo puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, el componente 221 de estimación de movimiento puede realizar una búsqueda de movimiento con respecto a las posiciones de píxeles completos y posiciones de píxeles fraccionarios y generar un vector de movimiento con precisión de píxeles fraccionarios. El componente 221 de estimación de movimiento calcula un vector de movimiento para una PU de un bloque de vídeo en un segmento intercodificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. El componente 221 de estimación de movimiento entrega el vector de movimiento calculado como datos de movimiento al componente 231 de formateo de cabecera y CABAC para la codificación y el movimiento para el componente 219 de compensación de movimiento.

La compensación de movimiento, realizada por el componente 219 de compensación de movimiento, puede implicar buscar o generar el bloque predictivo en base al vector de movimiento determinado por el componente 221 de estimación de movimiento. Nuevamente, el componente 221 de estimación de movimiento y el componente 219 de compensación de movimiento pueden integrarse funcionalmente, en algunos ejemplos. Al recibir el vector de movimiento para la PU del bloque de vídeo actual, el componente 219 de compensación de movimiento puede ubicar el bloque predictivo al que apunta el vector de movimiento. A continuación, se forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo de los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de

diferencia de píxel. En general, el componente 221 de estimación de movimiento realiza la estimación de movimiento con respecto a los componentes de luma, y el componente 219 de compensación de movimiento usa vectores de movimiento calculados en base a los componentes de luma tanto para los componentes de croma como para los componentes de luma. El bloque predictivo y el bloque residual se reenvían al componente 213 de escalado de transformada y cuantificación.

La señal de vídeo dividida 201 también se envía al componente 215 de estimación intraimagen y al componente 217 de predicción intraimagen. Al igual que con el componente 221 de estimación de movimiento y el componente 219 de compensación de movimiento, el componente 215 de estimación intraimagen y el componente 217 de predicción intraimagen pueden estar altamente integrados, pero se ilustran por separado para fines conceptuales. El componente 215 de estimación intraimagen y el componente 217 de predicción intraimagen intrapredicen un bloque actual con respecto a bloques en un cuadro actual, como una alternativa a la interpredicción realizada por el componente 221 de estimación de movimiento y el componente 219 de compensación de movimiento entre cuadros, como se describió anteriormente. En particular, el componente 215 de estimación intraimagen determina un modo de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, el componente 215 de estimación intraimagen selecciona un modo de intrapredicción apropiado para codificar un bloque actual a partir de múltiples modos de intrapredicción probados. Los modos de intrapredicción seleccionados se reenvían a continuación al componente 231 de formateo de cabecera y CABAC para su codificación.

Por ejemplo, el componente 215 de estimación intraimagen calcula valores de tasa-distorsión usando un análisis de tasa-distorsión para los varios modos de intrapredicción probados, y selecciona el modo de intrapredicción que tenga las mejores características de tasa-distorsión entre los modos probados. El análisis de tasa-distorsión determina, en general, una cantidad de distorsión (o error) entre un bloque codificado y un bloque original sin codificar que fue codificado para producir el bloque codificado, así como una tasa de bits (es decir, un número de bits) usada para producir el bloque codificado. El componente 215 de estimación intraimagen calcula relaciones a partir de las distorsiones y tasas para los varios bloques codificados para determinar qué modo de intrapredicción exhibe el mejor valor de tasa-distorsión para el bloque. Además, el componente 215 de estimación intraimagen puede configurarse para codificar bloques de profundidad de un mapa de profundidad usando un modo de modelado de profundidad (DMM) basado en optimización de tasa-distorsión (RDO).

El componente 217 de predicción intraimagen puede generar un bloque residual a partir del bloque predictivo en base a los modos de intrapredicción seleccionados determinados por el componente 215 de estimación intraimagen cuando se implementa en un codificador o leer el bloque residual del flujo de bits cuando se implementa en un decodificador. El bloque residual incluye la diferencia en valores entre el bloque predictivo y el bloque original, representado como una matriz. El bloque residual se reenvía entonces al componente 213 de escalado de transformada y cuantificación. El componente 215 de estimación intraimagen y el componente 217 de predicción intraimagen pueden funcionar tanto sobre componentes de luma como de croma.

El componente 213 de escalado de transformada y cuantificación está configurado para comprimir más el bloque residual. El componente 213 de escalado de transformada y cuantificación aplica una transformada, tal como una transformada de coseno discreta (DCT), una transformada de seno discreta (DST) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficiente de transformada residual. También podrían usarse transformadas de ondícula, transformadas de enteros, transformadas de subbanda u otros tipos de transformadas. La transformada puede convertir la información residual de un dominio de valor de píxel a un dominio de transformada, tal como un dominio de la frecuencia. El componente 213 de escalado de transformada y cuantificación también está configurado para escalar la información residual transformada, por ejemplo, en base a la frecuencia. Dicho escalado implica aplicar un factor de escala a la información residual para que la información de frecuencia diferente se cuantifique en diferentes granularidades, lo que puede afectar a la calidad visual final del vídeo reconstruido. El componente 213 de escalado de transformada y cuantificación también se configura para cuantificar los coeficientes de transformada para reducir más la tasa de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, el componente 213 de escalado de transformada y cuantificación puede realizar entonces un escaneo de la matriz que incluye los coeficientes de transformada cuantificados. Los coeficientes de transformada cuantificados se reenvían al componente 231 de formateo de cabecera y CABAC para codificarse en el flujo de bits.

El componente 229 de escalado y transformada inversa aplica una operación inversa del componente 213 de escalado de transformada y cuantificación para soportar la estimación de movimiento. El componente 229 de escalado y transformada inversa aplica escalado inverso, transformada y/o cuantificación para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso posterior como un bloque de referencia que puede convertirse en un bloque predictivo para otro bloque actual. El componente 221 de estimación de

movimiento y/o el componente 219 de compensación de movimiento pueden calcular un bloque de referencia añadiendo el bloque residual de nuevo a un bloque predictivo correspondiente para su uso en la estimación de movimiento de un bloque/cuadro posterior. Los filtros se aplican a los bloques de referencia reconstruidos para mitigar los artefactos creados durante el escalado, la cuantificación y la transformada. De lo contrario, dichos artefactos podrían causar una predicción inexacta (y crear artefactos adicionales) cuando se predicen bloques posteriores.

El componente 227 de análisis de control de filtros y el componente 225 de filtros en bucle aplican los filtros a los bloques residuales y/o a los bloques de imagen reconstruidos. Por ejemplo, el bloque residual transformado procedente del componente 229 de escalado y transformada inversa puede combinarse con un bloque de predicción correspondiente procedente del componente 217 de predicción intraimagen y/o del componente 219 de compensación de movimiento para reconstruir el bloque de imagen original. Los filtros pueden entonces aplicarse al bloque de imagen reconstruido. En algunos ejemplos, los filtros pueden aplicarse en cambio a los bloques residuales. Al igual que con otros componentes de la FIG. 2, el componente 227 de análisis de control de filtros y el componente 225 de filtros en bucle están altamente integrados y pueden implementarse juntos, pero se representan por separado con fines conceptuales. Los filtros aplicados a los bloques de referencia reconstruidos se aplican a regiones espaciales particulares e incluyen múltiples parámetros para ajustar cómo se aplican dichos filtros. El componente 227 de análisis de control de filtros analiza los bloques de referencia reconstruidos para determinar dónde se deben aplicar dichos filtros y ajusta los parámetros correspondientes. Dichos datos se reenvían al componente 231 de formateo de cabecera y CABAC como datos de control de filtros para su codificación. El componente 225 de filtros en bucle aplica dichos filtros en base a los datos de control de filtros. Los filtros pueden incluir un filtro de desbloqueo, un filtro de supresión de ruido, un filtro SAO y un filtro de bucle adaptativo. Dichos filtros pueden aplicarse en el dominio espacial/de píxeles (por ejemplo, en un bloque de píxeles reconstruido) o en el dominio de la frecuencia, dependiendo del ejemplo.

Cuando funciona como un codificador, el bloque de imagen reconstruido filtrado, el bloque residual y/o el bloque de predicción se almacenan en el componente 223 de memoria intermedia de imágenes decodificadas para su uso posterior en la estimación de movimiento como se analizó anteriormente. Cuando funciona como un decodificador, el componente 223 de memoria intermedia de imágenes decodificadas almacena y reenvía los bloques reconstruidos y filtrados hacia una pantalla como parte de una señal de vídeo de salida. El componente 223 de memoria intermedia de imágenes decodificadas puede ser cualquier dispositivo de memoria capaz de almacenar bloques de predicción, bloques residuales y/o bloques de imagen reconstruidos.

El componente 231 de formateo de cabecera y CABAC recibe los datos de los varios componentes del sistema de códec 200 y codifica dichos datos en un flujo de bits codificado para su transmisión hacia un decodificador. Específicamente, el componente 231 de formateo de cabecera y CABAC genera varias cabeceras para codificar datos de control, tales como datos de control general y datos de control de filtros. Además, los datos de predicción, incluidos los datos de intrapredicción y de movimiento, así como los datos residuales en la forma de datos de coeficientes de transformada cuantificados se codifican todos en el flujo de bits. El flujo de bits final incluye toda la información deseada por el decodificador para reconstruir la señal de vídeo dividida 201 original. Dicha información también puede incluir tablas de índice de modo de intrapredicción (también denominadas tablas de mapeo de palabras de código), definiciones de contextos de codificación para varios bloques, indicaciones de los modos de intrapredicción más probables, una indicación de información de división, etc. Dichos datos pueden codificarse empleando codificación entrópica. Por ejemplo, la información puede codificarse empleando codificación de longitud variable adaptable al contexto (CAVLC), (CABAC), codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), codificación entrópica de división en intervalos de probabilidad (PIPE) u otra técnica de codificación entrópica. Después de la codificación entrópica, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, un decodificador de vídeo) o archivar para su posterior transmisión o recuperación.

La FIG. 3 es un diagrama de bloques que ilustra un codificador de vídeo 300 de ejemplo. El codificador de vídeo 300 se puede emplear para implementar las funciones de codificación del sistema de códec 200 y/o implementar las etapas 101, 103, 105, 107 y/o 109 del método operativo 100. El codificador 300 divide una señal de vídeo de entrada, lo que tiene como resultado una señal de vídeo dividida 301, que es sustancialmente similar a la señal de vídeo dividida 201. A continuación, la señal de vídeo dividida 301 es comprimida y codificada en un flujo de bits por componentes del codificador 300.

Específicamente, la señal de vídeo dividida 301 se reenvía a un componente 317 de predicción intraimagen para la interpredicción. El componente 317 de predicción intraimagen puede ser sustancialmente similar al componente 215 de estimación intraimagen y al componente 217 de predicción intraimagen. La señal de vídeo dividida 301 también se reenvía a un componente 321 de compensación de movimiento para la interpredicción en base a bloques de referencia en un componente 323 de memoria intermedia de imágenes decodificadas. El componente 321 de compensación de movimiento puede ser sustancialmente similar al componente 221 de estimación de movimiento y al componente 219 de compensación de movimiento. Los

- bloques de predicción y los bloques residuales del componente 317 de predicción intraimagen y el componente 321 de compensación de movimiento se reenvían a un componente 313 de transformada y cuantificación para transformada y cuantificación de los bloques residuales. El componente 313 de transformada y cuantificación puede ser sustancialmente similar al componente 213 de escalado de transformada y cuantificación. Los bloques residuales transformados y cuantificados y los bloques de predicción correspondientes (junto con los datos de control asociados) se reenvían a un componente 331 de codificación entrópica para codificar en un flujo de bits. El componente 331 de codificación entrópica puede ser sustancialmente similar al componente 231 de formateo de cabecera y CABAC.
- 10 Los bloques residuales transformados y cuantificados y/o los bloques de predicción correspondientes también se reenvían desde el componente 313 de transformada y cuantificación a un componente 329 de transformada inversa y cuantificación para su reconstrucción en bloques de referencia para utilizar por el componente 321 de compensación de movimiento. El componente 329 de transformada inversa y cuantificación puede ser sustancialmente similar al componente 229 de escalado y transformada inversa. Los
- 15 filtros en bucle en un componente 325 de filtros en bucle también se aplican a los bloques residuales y/o bloques de referencia reconstruidos, dependiendo del ejemplo. El componente 325 de filtros en bucle puede ser sustancialmente similar al componente 227 de análisis de control de filtros y el componente 225 de filtros en bucle. El componente 325 de filtros en bucle puede incluir múltiples filtros como se ha explicado con respecto al componente 225 de filtros en bucle. Los bloques filtrados se almacenan entonces en un
- 20 componente 323 de memoria intermedia de imágenes decodificadas para su uso como bloques de referencia por el componente 321 de compensación de movimiento. El componente 323 de memoria intermedia de imágenes decodificadas puede ser sustancialmente similar al componente 223 de memoria intermedia de imágenes decodificadas.
- 25 La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 400. El decodificador de vídeo 400 puede ser empleado para implementar las funciones de decodificación del sistema de códec 200 y/o implementar las etapas 111, 113, 115 y/o 117 del método 100. El decodificador 400 recibe un flujo de bits, por ejemplo de un codificador 300, y genera una señal de vídeo de salida reconstruida en base al flujo de bits para mostrar a un usuario final.
- 30 El flujo de bits es recibido por un componente 433 de decodificación entrópica. El componente 433 de decodificación entrópica está configurado para implementar un esquema de decodificación entrópica, tal como CAVLC, CABAC, SBAC, codificación PIPE u otras técnicas de codificación entrópica. Por ejemplo, el componente 433 de decodificación entrópica puede emplear información de cabecera para proporcionar un
- 35 contexto para interpretar datos adicionales codificados como palabras de código en el flujo de bits. La información decodificada incluye cualquier información deseada para decodificar la señal de vídeo, tal como datos de control general, datos de control de filtros, información de división, datos de movimiento, datos de predicción y coeficientes de transformada cuantificados a partir de bloques residuales. Los coeficientes de transformada cuantificados se reenvían a un componente 429 de transformada inversa y cuantificación para su reconstrucción en bloques residuales. El componente 429 de transformada inversa y cuantificación puede ser similar al componente 329 de transformada inversa y cuantificación.
- 40 Los bloques residuales reconstruidos y/o los bloques de predicción se reenvían al componente 417 de predicción intraimagen para su reconstrucción en bloques de imagen en base a operaciones de intrapredicción. El componente 417 de predicción intraimagen puede ser sustancialmente similar al componente 215 de estimación intraimagen y a un componente 217 de predicción intraimagen. Específicamente, el componente 417 de predicción intraimagen emplea modos de predicción para localizar un bloque de referencia en un cuadro y aplica un bloque residual al resultado para reconstruir bloques de imagen intrapredichos. Los bloques de imagen intrapredichos reconstruidos y/o los bloques residuales y los
- 50 datos de interpredicción correspondientes se reenvían a un componente 423 de memoria intermedia de imágenes decodificadas por medio de un componente 425 de filtros en bucle, que puede ser sustancialmente similar al componente 223 de memoria intermedia de imágenes decodificadas y al componente 225 de filtros en bucle, respectivamente. El componente 425 de filtros en bucle filtra los bloques de imagen reconstruidos, bloques residuales y/o bloques de predicción, y dicha información se almacena en el componente 423 de memoria intermedia de imágenes decodificadas. Los bloques de imagen reconstruidos procedentes del
- 55 componente 423 de memoria intermedia de imágenes decodificadas se reenvían a un componente 421 de compensación de movimiento para la interpredicción. El componente 421 de compensación de movimiento puede ser sustancialmente similar al componente 221 de estimación de movimiento y al componente 219 de compensación de movimiento. Específicamente, el componente 421 de compensación de movimiento emplea vectores de movimiento de un bloque de referencia para generar un bloque de predicción y aplica un bloque residual al resultado para reconstruir un bloque de imagen. Los bloques reconstruidos resultantes también pueden ser reenviados por medio del componente 425 de filtros en bucle al componente 423 de memoria intermedia de imágenes decodificadas. El componente 423 de memoria intermedia de imágenes decodificadas continúa almacenando bloques de imagen reconstruidos adicionales, que se pueden
- 60 reconstruir en cuadros por medio de la información de división. Dichos cuadros también pueden colocarse en una secuencia. La secuencia es enviada hacia una pantalla como una señal de vídeo de salida reconstruida.
- 65

La FIG. 5 es un diagrama esquemático que ilustra un ejemplo de flujo de bits 500 que contiene una secuencia de vídeo codificada. Por ejemplo, el flujo de bits 500 puede ser generado por un sistema de códec 200 y/o un codificador 300 para decodificar por un sistema de códec 200 y/o un decodificador 400. Como otro ejemplo, el flujo de bits 500 puede ser generado por un codificador en la etapa 109 del método 100 para su uso por un decodificador en la etapa 111.

El flujo de bits 500 incluye un conjunto de parámetros de secuencia (SPS) 510, una pluralidad de conjuntos de parámetros de imagen (PPS) 512, cabeceras de segmento 514 y datos de imagen 520. Un SPS 510 contiene datos de secuencia comunes a todas las imágenes en la secuencia de vídeo contenida en el flujo de bits 500. Dichos datos pueden incluir el tamaño de la imagen, la profundidad de bits, los parámetros de la herramienta de codificación, las restricciones de velocidad de bits, etc. El PPS 512 contiene parámetros que son específicos para una o más imágenes correspondientes. Por lo tanto, cada imagen en una secuencia de vídeo puede referirse a un PPS 512. El PPS 512 puede indicar herramientas de codificación disponibles para segmentos (o teselas) en imágenes correspondientes, parámetros de cuantificación, desplazamientos, parámetros de herramientas de codificación específicos de imágenes (por ejemplo, controles de filtros), etc. La cabecera de segmento 514 contiene parámetros que son específicos para cada segmento de una imagen. Por lo tanto, puede haber una cabecera de segmento 514 por segmento en la secuencia de vídeo. La cabecera de segmento 514 puede contener información de segmento, cómputos de orden de imagen (POC), listas de imágenes de referencia, ponderaciones de predicción, puntos de entrada de tesela, parámetros de desbloqueo, etc. Cabe señalar que algunos sistemas se refieren a la cabecera de segmento 514 como una cabecera de grupo de teselas, y utilizan dicha información para soportar grupos de teselas en lugar de segmentos.

Los datos de imagen 520 contienen datos de vídeo codificados de acuerdo con inter-predicción y/o intra-predicción, así como los datos residuales transformados y cuantificados correspondientes. Dichos datos de imagen 520 se clasifican de acuerdo con la división utilizada para dividir la imagen antes de la codificación. Por ejemplo, la imagen en los datos de imagen 520 incluye una o más imágenes 521. La secuencia o serie de imágenes 521 puede denominarse CVS 527. Tal como se utiliza en el presente documento, la CVS 527 es una secuencia de unidades de acceso (AU) que incluye, en orden de decodificación, una AU de inicio de secuencia de vídeo codificada (CVSS), seguida de cero o más AU que no son AU CVSS, incluidas todas las AU posteriores, pero sin incluir ninguna AU posterior que sea una AU CVSS. Una AU CVSS es una AU en la que hay una unidad de predicción (PU) para cada capa especificada por el conjunto de parámetros de vídeo (VPS) y la imagen codificada en cada PU es una imagen de inicio de secuencia de vídeo de capa codificada (CLVSS). En una realización, cada imagen 521 se dispone dentro de una AU. Una PU es un conjunto de NAL que están asociadas entre sí de acuerdo con una regla de clasificación especificada, son consecutivas en orden de decodificación y contienen exactamente una imagen codificada.

La CVS es una secuencia de vídeo codificada para cada secuencia de vídeo de capa codificada (CLVS) en el flujo de bits. En particular, la CVS y la CLVS son las mismas cuando el flujo de bits de vídeo incluye una sola capa. La CVS y la CLVS solo son diferentes cuando el flujo de bits de vídeo incluye múltiples capas.

Cada una de las imágenes 521 se puede dividir en subimágenes 523. Una subimagen 523 es una porción dividida de una imagen creada por límites horizontales y verticales. Las subimágenes 523 pueden ser rectangulares y/o cuadradas. Específicamente, una subimagen 523 incluye cuatro lados que están conectados en ángulos rectos. Los cuatro lados incluyen dos pares de lados paralelos. Además, los lados en un par lateral paralelo son de igual longitud. Como tal, una subimagen 523 puede ser cualquier forma rectangular, donde un cuadrado es un caso especial de un rectángulo donde los cuatro lados tienen la misma longitud.

Cada subimagen 523 se puede dividir en segmentos 529. Cabe señalar que, en algunas aplicaciones, los segmentos 529 pueden denominarse grupos de teselas y/o teselas. Los segmentos 529 pueden ser rectangulares y/o cuadrados, lo que significa que los segmentos tienen una forma de rectángulo o cuadrado (a diferencia de, por ejemplo, los segmentos de barrido de trama que pueden no tener una forma rectangular o cuadrada). Específicamente, un segmento 529 incluye cuatro lados que están conectados en ángulos rectos. Los cuatro lados incluyen dos pares de lados paralelos. Además, los lados en un par lateral paralelo son de igual longitud. Como tal, un segmento 529 puede ser cualquier forma rectangular, donde un cuadrado es un caso especial de un rectángulo donde los cuatro lados tienen la misma longitud.

En una realización, cada subimagen 523 comprende una región rectangular de uno o más segmentos 529 dentro de una imagen 521. Es decir, cada subimagen 523 se divide en uno o más segmentos 529, y cada imagen 521 se divide en una o más subimágenes 523.

Los segmentos 529 pueden o no permitir predicción basada en otros segmentos 529, dependiendo del ejemplo. Cada segmento 529 puede tener un índice de segmento único en la subimagen 523. Un índice de segmento es un identificador numérico seleccionado por procedimiento que se puede utilizar para distinguir

un segmento 529 de otra. Por ejemplo, los índices de segmento pueden aumentar numéricamente de arriba a abajo o de izquierda a derecha.

5 Cabe señalar que, en algunos ejemplos, también se pueden asignar identificadores (ID) de segmento a los segmentos 529. Un ID de segmento es un identificador asignado que se puede utilizar para distinguir un segmento 529 de otro. En algunos ejemplos, los cálculos pueden emplear ID de segmentos en lugar de índices de segmentos. Además, se pueden asignar ID de segmento para que tengan los mismos valores que los índices de segmento en algunos ejemplos. Los índices de segmentos y/o ID pueden ser señalizados para
10 indicar grupos de segmentos que contienen los segmentos 529. Por ejemplo, los índices de segmento y/o ID pueden ser empleados para mapear los datos de imagen asociados con un segmento 529 a una posición apropiada para visualización. Una pluralidad de segmentos 529 puede denominarse un conjunto de teselas. Un conjunto de teselas es un conjunto relacionado de segmentos 529 que se pueden extraer y codificar por separado, por ejemplo, para soportar la visualización de una región de interés y/o para soportar procesamiento paralelo. Los segmentos 529 en un conjunto de teselas se pueden codificar sin referencia a segmentos 529 fuera del grupo de teselas. Cada segmento 529 puede asignarse a un conjunto de teselas correspondiente y, por lo tanto, una imagen 521 puede contener una pluralidad de conjuntos de teselas.

Los segmentos 529 se dividen a su vez en unidades de árbol de codificación (CTU). Las CTU se dividen a su vez en bloques de codificación en base a árboles de codificación. Los bloques de codificación se pueden, a
20 continuación, codificar/decodificar de acuerdo con los mecanismos de predicción.

En una realización, el PPS 512 contiene un indicador de segmento único por subimagen 532. En una realización, el indicador de segmento único por subimagen 532 tiene un primer valor (por ejemplo, uno). El primer valor especifica que cada subimagen (por ejemplo, la subimagen 523) que se refiere al PPS 512 (a diferencia de otro PPS en el flujo de bits 500) puede incluir uno y solo un segmento rectangular (por ejemplo, el segmento 529). Es decir, cada subimagen incluye un único segmento rectangular. Por lo tanto, las dimensiones de la subimagen y el segmento rectangular son las mismas.
25

En una realización, el indicador de segmento único por subimagen 532 tiene un segundo valor (por ejemplo, cero). El segundo valor especifica que cada subimagen (por ejemplo, la subimagen 523) que se refiere al PPS 512 (a diferencia de otro PPS en el flujo de bits 500) puede incluir uno o más segmentos rectangulares (por ejemplo, segmentos 529). Es decir, cada subimagen puede incluir un único segmento rectangular o puede contener una pluralidad de los segmentos. Cuando una subimagen incluye una pluralidad de los segmentos, las dimensiones de la subimagen son diferentes de las dimensiones de los segmentos individuales. En una realización, el indicador de segmento único por subimagen 532 se designa como `pps_single_slice_per_subpic_flag`.
30
35

En una realización, el PPS 512 además comprende un indicador del número de segmentos en imagen 534. El indicador del número de segmentos en imagen 534 se utiliza para determinar el número de segmentos rectangulares (por ejemplo, segmentos 529) en cada imagen (por ejemplo, 521) que se refiere al PPS 512. En una realización, el indicador del número de segmentos en imagen 534 se utiliza para significar el número de segmentos rectangulares en cada imagen que se refiere al PPS cuando el indicador de segmento único por subimagen 532 tiene el segundo valor.
40

En una realización, un valor del indicador del número de segmentos en imagen 534, más uno, especifica el número de segmentos en cada imagen que se refiere al PPS. Por ejemplo, cuando el valor del indicador del número de segmentos en imagen 534 es tres, el número total de segmentos en la imagen es cuatro (por ejemplo, tres + uno). En una realización, el indicador del número de segmentos en imagen 534 se designa como `pps_num_slices_in_pic_minus1`.
45
50

La FIG. 6 es una realización de un método 600 de decodificación implementado por un decodificador de vídeo (por ejemplo, el decodificador de vídeo 400). El método 600 puede realizarse después de que el flujo de bits decodificado haya sido recibido directa o indirectamente desde un codificador de vídeo (por ejemplo, el codificador de vídeo 300). El método 600 mejora el proceso de decodificación mediante el uso de un
55 indicador en el flujo de bits para indicar si cada subimagen de una imagen contiene un, y solo un, segmento rectangular. Cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular, los elementos de sintaxis para señalar el número y la distribución de las subimágenes (por ejemplo, la posición y el tamaño de cada subimagen) se pueden omitir del flujo de bits porque las subimágenes son las mismas que los segmentos. Es decir, solo los elementos de sintaxis para los segmentos tienen que incluirse en el flujo de bits cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular. Esto reduce la cantidad de información transportada en el flujo de bits, reduce la redundancia y aumenta la eficiencia de la codificación. Por lo tanto, el codificador/decodificador (también conocido como "códec") en codificación de vídeo se mejora en relación con los códecs actuales. Como cuestión práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia de usuario cuando se
60 envían, reciben y/o ven vídeos.
65

- En el bloque 602, el decodificador de vídeo recibe un flujo de bits (por ejemplo, flujo de bits 500) que incluye un PPS (por ejemplo, PPS 512) que contiene un indicador de segmento único por subimagen (por ejemplo, indicador de segmento único por subimagen 532). En una realización, el indicador de segmento único por subimagen se designa como `pps_single_slice_per_subpic_flag`.
- 5 En una realización, el PPS además comprende un indicador del número de segmentos en imagen (por ejemplo, indicador del número de segmentos en imagen 534). Un valor del indicador del número de segmentos en imagen más uno especifica un número de segmentos en cada imagen que se refiere al PPS. En una realización, el indicador del número de segmentos en imagen se designa como `pps_num_slices_in_pic_minus1`.
- 10 En el bloque 604, el decodificador de vídeo determina si el indicador de segmento único por subimagen tiene un primer valor o un segundo valor. El primer valor especifica que cada subimagen que se refiere al PPS incluye uno y sólo un segmento rectangular. El segundo valor especifica que cada subimagen que se refiere al PPS puede incluir uno o más segmentos rectangulares (por ejemplo, una o una pluralidad de segmentos rectangulares). En una realización, el primer valor es uno (1) y el segundo valor es cero (0). En una realización, cada subimagen comprende una región rectangular de uno o más segmentos dentro de una imagen.
- 15 En el bloque 606, el decodificador de vídeo decodifica el uno y sólo un segmento rectangular cuando el indicador de segmento único por subimagen tiene el primer valor y el uno o más segmentos rectangulares cuando el indicador de segmento único por subimagen tiene el segundo valor. Una vez decodificado, el uno y solo un segmento rectangular o el uno o más segmentos rectangulares, según el caso, se utilizan para generar o producir una secuencia de imagen o de vídeo para mostrarla a un usuario en la pantalla de un dispositivo electrónico (por ejemplo, un teléfono inteligente, tableta, ordenador portátil, ordenador personal, etc.).
- 20 La FIG. 7 es una realización de un método 700 de codificación de un flujo de bits de vídeo implementado por un codificador de vídeo (por ejemplo, el codificador de vídeo 300). El método 700 puede realizarse cuando una imagen (por ejemplo, de un vídeo) va a codificarse en un flujo de bits de vídeo y luego transmitirse hacia un decodificador de vídeo (por ejemplo, el decodificador de vídeo 400). El método 700 mejora el proceso de codificación mediante el uso de un indicador en el flujo de bits para indicar si cada subimagen de una imagen contiene un, y solo un, segmento rectangular. Cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular, los elementos de sintaxis para señalar el número y la distribución de las subimágenes (por ejemplo, la posición y el tamaño de cada subimagen) se pueden omitir del flujo de bits porque las subimágenes son las mismas que los segmentos. Es decir, solo los elementos de sintaxis para los segmentos tienen que incluirse en el flujo de bits cuando el indicador indica que cada subimagen contiene solo un único segmento rectangular. Esto reduce la cantidad de información transportada en el flujo de bits, reduce la redundancia y aumenta la eficiencia de la codificación. Por lo tanto, el codificador/decodificador (también conocido como "códec") en codificación de vídeo se mejora en relación con los códecs actuales. Como cuestión práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia de usuario cuando se envían, reciben y/o ven vídeos.
- 30 En el bloque 702, el codificador de vídeo establece un indicador de segmento único por subimagen (por ejemplo, un indicador de segmento único por subimagen 532) en un primer valor cuando cada subimagen (por ejemplo, subimagen 523) que se refiere a un PPS (por ejemplo, PPS 512) incluye uno y solo un segmento rectangular y un segundo valor cuando cada subimagen que se refiere al PPS puede incluir uno o más segmentos rectangulares.
- 35 En una realización, el indicador de segmento único por subimagen se designa como `pps_single_slice_per_subpic_flag`. En una realización, el primer valor es uno (1) y el segundo valor es cero (0).
- 40 En una realización, el PPS además comprende un indicador del número de segmentos en imagen (por ejemplo, indicador del número de segmentos en imagen 534). En una realización, el codificador de vídeo ajusta el indicador del número de segmentos en imagen a un valor. El valor del indicador del número de segmentos en imagen más uno especifica un número de segmentos en cada imagen que se refiere al PPS. En una realización, el indicador del número de segmentos en imagen se designa como `pps_num_slices_in_pic_minus1`. En una realización, cada subimagen comprende una región rectangular de uno o más segmentos dentro de una imagen.
- 45 En el bloque 704, el codificador de vídeo codifica el PPS que contiene el indicador de segmento único por subimagen en un flujo de bits (por ejemplo, flujo de bits 500). En el bloque 706, el codificador de vídeo almacena el flujo de bits para su comunicación hacia un decodificador. El flujo de bits puede ser almacenado en memoria hasta que el flujo de bits de vídeo es transmitido hacia el decodificador de vídeo. Una vez recibido por el decodificador de vídeo, el flujo de bits codificado puede ser decodificado (por ejemplo, como
- 50
- 55
- 60
- 65

se describe anteriormente) para generar o producir una secuencia de imagen o de vídeo para mostrar a un usuario en la pantalla de un dispositivo electrónico (por ejemplo, un teléfono inteligente, tableta, ordenador portátil, ordenador personal, etc.).

5 La siguiente sintaxis y semántica pueden emplearse para implementar las realizaciones dadas a conocer en la presente memoria. La siguiente descripción se relaciona con el texto base, que es la última especificación de borrador de VVC, suponiendo que la señalización de la información de la subimagen se realiza en función de la distribución de los segmentos. En otras palabras, solo se describe el delta, si bien el texto en el texto base que no se menciona a continuación se aplica tal cual.

10

La sintaxis de PPS es la siguiente.

	Descriptor
pic_parameter_set_rbsp () {	
..	
single_tile_in_pic_flag	u(1)
if(!single_tile_in_pic_flag) {	
...	
loop_filter_across_bricks_enabled_flag	u(1)
if(loop_filter_across_bricks_enabled_flag)	
loop_filter_across_slices_enabled_flag	u(1)
}	
if(rect_slice_flag) {	
subpics_present_flag	u(1)
if(subpics_present_flag) {	
if(num_slices_in_pic_minus1 > 0)	
single_slice_per_subpic_flag	u(1)
if(!single_slice_per_subpic_flag)	
num_subpics_minus1	UE (v)
if(num_subpics_minus1 > 0)	
single_slice_per_subpic_flag	
bottom_right_slice_idx_length_minus1	UE (v)
for(i = 0; i < num_subpics_minus1; i++) {	
bottom_right_slice_idx_delta[i]	U (V)
si(i>0)	
slice_idx_delta_sign_flag[i]	u(1)
}	
}	
for(i = 0; i <= num_subpics_minus1; i++) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
subpic_ids_constant_in_clvs_flag	u(1)
subpic_id_len_minus1	UE (v)
for(i = 0; i <= num_subpics_minus1; i++)	
pps_subpic_id[i]	U (V)
}	
}	
entropy_coding_sync_enabled_flag	u(1)
..	
rbps_bits_finales()	
}	

La semántica de PPS es la siguiente.

15

brick_idx_delta_sign_flag[i] igual a 1 indica un signo positivo para **bottom_right_brick_idx_delta[i]**. **brick_idx_delta_sign_flag[i]** igual a 0 indica un signo negativo para **bottom_right_brick_idx_delta[i]**.

20

Es un requisito de conformidad del flujo de bits que un segmento debe incluir ya sea un número de teselas completas o solo una secuencia consecutiva de ladrillos completos de una tesela.

La variable **TopLeftBrickIdx[i]**, **BottomRightBrickIdx[i]**, **NumBricksInSlice[i]**, **BrickToSliceMap[j]**, **SliceColBd[i]**, **SliceRowBd[i]**, **SliceWidth[i]** y **SliceHeight[i]**, que especifican el índice de ladrillos del ladrillo ubicado en la esquina superior izquierda del i-ésimo segmento, el índice de ladrillos del ladrillo

- ubicado en la esquina inferior derecha del *i*-ésimo segmento, el número de ladrillos en el *i*-ésimo segmento, y el índice de segmento del segmento que contiene el *j*-ésimo ladrillo, la ubicación del límite izquierdo del *i*-ésimo segmento en unidades de CTB, la ubicación del límite superior del *i*-ésimo segmento en unidades de CTB, la anchura del *i*-ésimo segmento en unidades de CTB y la altura del *i*-ésimo segmento en unidades de CTB, respectivamente, se obtienen de la siguiente manera:

```

5
for( j = 0; i == 0 && j < NumBricksInPic; j++ )
    BrickToSliceMap[ j ] = -1
NumBricksInSlice[ i ] = 0
if( i == num_slices_in_pic_minus1 )
    BottomRightBrickIdx[ i ] = NumBricksInPic - 1
else
    BottomRightBrickIdx[ i ] = i == 0 ? bottom_right_brick_idx_delta[ i ] :
        ( BottomRightBrickIdx[ i - 1 ] + ( brick_idx_delta_sign_flag[ i ] ?
            bottom_right_brick_idx_delta[ i ] : -bottom_right_brick_idx_delta[ i ] ) )
for( j = BottomRightBrickIdx[ i ]; j >= 0; j-- ) {
    if( BrickColBd[ j ] + BrickWidth[ j ] <= BrickColBd[ BottomRightBrickIdx[ i ] ] +
        BrickWidth[ BottomRightBrickIdx[ i ] ] &&
        BrickRowBd[ j ] + BrickHeight[ j ] <= BrickRowBd[ BottomRightBrickIdx[ i ] ] +
        BrickHeight[ BottomRightBrickIdx[ i ] ] &&
        BrickToSliceMap[ j ] == -1 ) {
        TopLeftBrickIdx[ i ] = j
        NumBricksInSlice[ i ]++
        BrickToSliceMap[ j ] = i
    }
}
SliceColBd[ i ] = BrickColBd[ TopLeftBrickIdx[ i ] ]
SliceRowBd[ i ] = BrickRowBd[ TopLeftBrickIdx[ i ] ]
SliceWidth[ i ] = BrickColBd[ BottomRightBrickIdx[ i ] ] +
    BrickWidth[ BottomRightBrickIdx[ i ] ] - SliceColBd[ i ]
SliceHeight[ i ] = BrickRowBd[ BottomRightBrickIdx[ i ] ] +
    BrickHeight[ BottomRightBrickIdx[ i ] ] - SliceRowBd[ i ]

```

- 10 **subpics_present_flag** igual a 1 indica que los parámetros de subimagen están presentes en la sintaxis RBSP de PPS. **subpics_present_flag** igual a 0 indica que los parámetros de subimagen no están presentes en la sintaxis RBSP de PPS. Cuando no está presente, el valor del indicador **subpics_present** se deduce que es igual a 0.
- 15 NOTA - Cuando un flujo de bits es el resultado de un proceso de extracción de subflujo de bits basado en subimágenes en el que el flujo de bits de entrada contiene múltiples subimágenes por imagen y durante la extracción las unidades NAL de VCL solo pueden extraerse o desecharse, pero no cambiarse, y el flujo de bits contiene solo un subconjunto verdadero de las subimágenes del flujo de bits de entrada, el valor del indicador **subpics_present** tiene que ser igual a 1 en la RBSP de los PPS en el flujo de bits, incluso cuando
- 20 solo hay una subimagen en cada imagen.

single_slice_per_subpic_flag igual a 1 especifica que cada subimagen en cada imagen que se refiere a este PPS incluye un segmento rectangular. **single_slice_per_subpic_flag** igual a 0 especifica que una subimagen en una imagen que se refiere a este PPS puede incluir más de un segmento rectangular. Cuando

el indicador `rect slice` es igual a 1 y `num slices in_pic` menos 1 es igual a 0, el valor de `single_slice_per_subpic_flag` se deduce que es igual a 1.

5 **num_subpics_minus1** más 1 especifica el número de subimágenes en cada imagen que se refiere al PPS. `num_subpics_minus1` estará en el intervalo de 0 a `num slices in_pic` menos 1, inclusive. Cuando el indicador `subpics_present` es igual a 0, el valor de `num subimágenes menos 1` se deduce a 0. Cuando no está presente y `single_slice_per_subpic_flag` es igual a 1, el valor de `num_subpics_minus1` se deduce que es igual a `num_slices_in_pic_minus1`. La variable `NumSubPics` se deduce que es igual a `num_subpics_minus1 + 1`.

10 **bottom_right_slice_idx_length_minus1** más 1 especifica el número de bits utilizados para representar el elemento de sintaxis `bottom_right_slice_idx_delta[i]`. El valor del segmento inferior derecho `idx_length_minus1` estará en el intervalo de 0 a $\text{Techo}(\text{Log2}(\text{num_slices_in_pic_minus1} + 1)) - 1$, inclusive.

15 **bottom_right_slice_idx_delta[i]** cuando `i` es mayor que 0 especifica la diferencia entre el índice de segmento del segmento ubicado en la esquina inferior derecha de la `i`-ésima subimagen y el índice de segmento de la esquina inferior derecha de la `(i - 1)`-ésima subimagen. `bottom_right_slice_idx_delta[0]` especifica el índice de segmento de la esquina inferior derecha de la 0-ésima subimagen. Cuando `single_slice_per_subpicflag` es igual a 1, el valor de `bottom_right_slice_idx_delta[i]` se deduce que es igual a 1. El valor de `BottomRightSliceIdx[num_subpics_minus1]` se deduce que es igual a `num_slices_in_pic_minus1`. La longitud del elemento de sintaxis `bottom_right_slice_idx_delta[i]` es `bottom_right_slice_idx_length_minus1 + 1` bits.

20 **slice_idx_delta_sign_flag[i]** igual a 1 indica un signo positivo para `bottom_right_slice_idx_delta[i]`. `slice_idx_delta_sign_flag[i]` igual a 0 indica un signo negativo para `bottom_right_slice_idx_delta[i]`.

25 La variable `TopLeftSliceIdx[i]`, `BottomRightSliceIdx[i]`, `NumSlicesInSubpic[i]`, `SliceToSubpicMap[j]`, y `SliceSubpicToPicIdx[i][k]`, que especifican el índice de segmento del segmento ubicado en la esquina superior izquierda de la `i`-ésima subimagen, el índice de segmento del segmento ubicado en la esquina inferior derecha de la `i`-ésima subimagen, el número de segmentos en la `i`-ésima subimagen, el índice de subimagen de la subimagen que contiene el `j`-ésimo segmento, y el índice de segmento de nivel de imagen del `k`-ésimo segmento en la `i`-ésima subimagen, respectivamente, se obtienen de la siguiente manera.

30

```

if( single_slice_per_subpic_flag ) {
    TopLeftSliceIdx[ i ] = i
    BottomRightSliceIdx[ i ] = i
    NumSlicesInSubpic[ i ] = 1

```

```

SliceToSubpicMap[ i ] = i
SliceSubpicToPicIdx[ i ][ 0 ] = 0
} else {
for( j = 0; i == 0 && j <= num_slices_in_pic_minus1; j++ )
    SliceToSubpicMap[ j ] = -1
NumSlicesInSubpic[ i ] = 0
if( i == num_subpics_minus1 )
    BottomRightSliceIdx[ i ] = num_slices_in_pic_minus1
else
    BottomRightSliceIdx[ i ] = i == 0 ? bottom_right_slice_idx_delta[ i ] :
        ( BottomRightSliceIdx[ i-1 ] + ( slice_idx_delta_sign_flag[ i ] ?
            bottom_right_slice_idx_delta[ i ] : -bottom_right_slice_idx_delta[ i ] ) )
for( j = BottomRightSliceIdx[ i ]; j >= 0; j-- ) {
    if( SliceColBd[ j ] + SliceWidth[ j ] <=
        SliceColBd[ BottomRightSliceIdx[ i ] ] + SliceWidth[ BottomRightSliceIdx[ i ] ] &&
        SliceRowBd[ j ] + SliceHeight[ j ] <=
        SliceRowBd[ BottomRightSliceIdx[ i ] ] + SliceHeight[ BottomRightSliceIdx[ i ] ] &&
        SliceToSubpicMap[ j ] == -1 ) {
        TopLeftSliceIdx[ i ] = j
        NumSlicesInSubpic[ i ]++
        SliceToSubpicMap[ j ] = i
    }
}
for( j = 0, k = 0; j < num_slices_in_pic_minus1 && k < NumSlicesInSubpic[ i ]; j++ )
    if( SliceToSubpicMap[ j ] == i )
        SliceSubpicToPicIdx[ i ][ k++ ] = j
}

```

- Las variables `SubpicLeft[i]`, `SubpicTop[i]`, `SubpicWidth` y `SubpicHeight[i]`, que especifican la posición del límite izquierdo, la posición del límite superior, la anchura y la altura, respectivamente, de la *i*-ésima subimagen en unidad de CTB se obtienen de la siguiente manera para cada valor de *i* en el intervalo de 0 a `NumSubPics - 1`, inclusive.

```

SubpicLeft[ i ] = SliceColBd[ TopLeftSliceIdx[ i ] ]
SubpicWidth[ i ] = SliceColBd[ BottomRightSliceIdx[ i ] ] +
    SliceWidth[ BottomRightSliceIdx[ i ] ] - SubpicLeft[ i ]
SubpicTop[ i ] = SliceRowBd[ TopLeftSliceIdx[ i ] ]
SubpicHeight[ i ] = SliceRowBd[ BottomRightSliceIdx[ i ] ] +
    SliceHeight[ BottomRightSliceIdx[ i ] ] - SubpicTop[ i ]

```

- 10 **subpic_treated_as_pic_flag[i]** igual a 1 especifica que la *i*-ésima subimagen de cada imagen codificada en la CVS se trata como una imagen en el proceso de decodificación excluyendo las operaciones de filtrado en bucle. **subpic_treated_as_pic_flag[i]** igual a 0 especifica que la *i*-ésima subimagen de cada imagen codificada en la CVS no se trata como una imagen en el proceso de decodificación excluyendo las operaciones de filtrado en bucle. Cuando no está presente, el valor de `subpic_treated_as_pic_flag[i]` se deduce que es igual a 0.

loop_filter_across_subpic_enabled_flag[i] igual a 1 especifica que las operaciones de filtrado en bucle se pueden realizar a través de los límites de la i-ésima subimagen en cada imagen codificada en la CVS. **loop_filter_across_subpic_enabled_flag**[i] igual a 0 especifica que las operaciones de filtrado en bucle no se realizan a través de los límites de la i-ésima subimagen en cada imagen codificada en la CVS. Cuando no está presente, el valor de **loop_filter_across_subpic_enabled_pici**] se deduce que es igual a 1.

Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones.

- Para dos subimágenes subpicA y subpicB cualesquiera, cuando el índice de subpicA sea menor que el índice de subpicB, cualquier unidad NAL codificada de subPicA sucederá a cualquier unidad NAL codificada de subPicB en orden de decodificación.

Las formas de las subimágenes serán tales que cada subimagen, cuando esté decodificada, tendrá todo su límite izquierdo y todo su límite superior consistiendo en límites de imágenes o consistiendo en límites de subimágenes previamente decodificadas.

Los valores de **SubpicLeft**[i], **SubpicTop**[i], **SubpicWidth**[i] y **SubpicHeight**[i] serán los mismos, respectivamente, para cada valor de i en el intervalo 0 a NumSubPics - 1, inclusive, independientemente de la RBSP PPS a la que se haga referencia por las unidades NAL de segmento codificado de una CLVS de la que se deriven.

Los valores de **subpic_treated_as_pic_flag**[i] y **loop_filter_across_subpic_enabled_flag**[i] seguirán siendo los mismos en todas las RBSP PPS a las que se hace referencia por las unidades NAL de segmento codificado de una CLVS respectivamente para cada valor de i en el intervalo de 0 a NumSubPics - 1, inclusive.

subpic_ids_constant_in_clvs_flag igual a 1 indica que el id de subpic len minus1 y los valores de **pps_subpic_id**[i], para cada valor de i en el intervalo de 0 a num subpics_minus1, inclusive, siguen siendo los mismos en todos las RBSP de PPS a las que se hace referencia por las unidades NAL de segmento codificado de una CLVS. **subpic_ids_constant_in_clvs_flag** igual a 0 indica que los valores de **subpic_id_len_minus1** y los valores de **pps_subpic_id**[i] pueden o no estar restringidos.

subpic_id_len_minus1 más 1 especifica la longitud de los elementos de sintaxis **pps_subpic_id**[i] y **slice_subpic_id** en bits. El valor de **subpic_id_len_minus1** estará comprendido en el intervalo de 3 a 31, ambos inclusive. Los valores de la **subpic_id_len_minus1** en el intervalo de 0 a 2, inclusive, están reservados para uso futuro por la UIT-T | ISO/CEI.

pps_subpic_id[i] especifica el identificador de la i-ésima subimagen.

La FIG. 8 es un diagrama esquemático de un dispositivo de codificación de vídeo 800 (por ejemplo, un codificador de vídeo 300 o un decodificador de vídeo 400) según una realización de la descripción. El dispositivo de codificación de vídeo 800 es adecuado para implementar las realizaciones dadas a conocer como se describe en la presente memoria. El dispositivo de codificación de vídeo 800 comprende puertos 810 de entrada y unidades de recepción (Rx) 820 para recibir datos; un procesador, unidad lógica o unidad central de procesamiento (CPU) 830 para procesar los datos; unidades transmisoras (Tx) 840 y puertos 850 de salida para transmitir los datos; y una memoria 860 para almacenar los datos. El dispositivo de codificación de vídeo 800 también puede comprender componentes ópticos a eléctricos (OE) y componentes eléctricos a ópticos (EO) acoplados a los puertos 810 de entrada, a las unidades 820 de recepción, a las unidades 840 de transmisión y a los puertos 850 de salida para la salida o entrada de señales ópticas o eléctricas.

El procesador 830 está implementado por hardware y software. El procesador 830 puede implementarse como uno o más chips de CPU, núcleos (por ejemplo, como un procesador multinúcleo), matrices de puertas programable en campo (FPGA), circuitos integrados de aplicación específica (ASIC) y procesadores de señales digitales (DSP). El procesador 830 está en comunicación con el puerto de entrada 810, la unidad 820 de recepción, las unidades 840 de transmisión, los puertos 850 de salida y la memoria 860. El procesador 830 comprende un módulo 870 de codificación. El módulo 870 de codificación implementa las realizaciones dadas a conocer descritas anteriormente. Por ejemplo, el módulo 870 de codificación implementa, procesa, prepara o proporciona las diversas funciones de códec. La inclusión del módulo 870 de codificación por lo tanto proporciona una mejora sustancial a la funcionalidad del dispositivo de codificación de vídeo 800 y efectúa una transformación del dispositivo de codificación de vídeo 800 a un estado diferente. Alternativamente, el módulo 870 de codificación se implementa como instrucciones almacenadas en la memoria 860 y ejecutadas por el procesador 830.

ES 3 010 349 T3

5 El dispositivo de codificación de vídeo 800 también puede incluir dispositivos 880 de entrada y/o salida (I/O) para comunicar datos hacia y desde un usuario. Los dispositivos 880 de I/O pueden incluir dispositivos de salida, tales como una pantalla para mostrar datos de vídeo, altavoces para emitir datos de audio, etc. Los dispositivos 880 de I/O también pueden incluir dispositivos de entrada, tales como un teclado, ratón, rueda de desplazamiento, etc. , y/o interfaces correspondientes para interactuar con tales dispositivos de salida.

10 La memoria 860 comprende uno o más discos, unidades de cinta, y/o unidades de estado sólido y puede ser usada como un dispositivo de almacenamiento de datos de desbordamiento para almacenar programas cuando tales programas se seleccionan para ejecutarlos, y para almacenar instrucciones y datos que se leen durante la ejecución del programa. La memoria 860 puede ser volátil y/o no volátil y puede ser memoria de solo lectura (ROM), memoria de acceso aleatorio (RAM), memoria ternaria de contenido direccionable (TCAM), y/o memoria estática de acceso aleatorio (SRAM).

15 La FIG. 9 es un diagrama esquemático de la realización de un medio de codificación 900. En una realización, el medio de codificación 900 se implementan en un dispositivo de codificación de vídeo 902 (por ejemplo, un codificador de vídeo 300 o un decodificador de vídeo 400). El dispositivo de codificación de vídeo 902 incluye medios 901 de recepción. Los medios 901 de recepción están configurados para recibir una imagen para codificar o para recibir un flujo de bits a decodificar. El dispositivo de codificación de vídeo 902 incluye medios 907 de transmisión acoplados a los medios 901 de recepción. Los medios 907 de transmisión están configurados para transmitir el flujo de bits a un decodificador o para transmitir una imagen decodificada a un medio de visualización (por ejemplo, uno de los dispositivos 880 de I/O).

25 El dispositivo de codificación de vídeo 902 incluye unos medios 903 de almacenamiento. Los medios 903 de almacenamiento están acoplados a al menos uno de los medios 901 de recepción o a los medios 907 de transmisión. Los medios 903 de almacenamiento están configurados para almacenar instrucciones. El dispositivo de codificación de vídeo 902 también incluye medios 905 de procesamiento. Los medios 905 de procesamiento están acoplados a los medios 903 de almacenamiento. Los medios 905 de procesamiento están configurados para ejecutar las instrucciones almacenadas en los medios 903 de almacenamiento para realizar los métodos dados a conocer en la presente memoria.

30

REIVINDICACIONES

1. Un método implementado por un decodificador, que comprende:
- 5 recibir (602), por el decodificador, un flujo de bits que incluye un conjunto de parámetros de imagen, PPS, que incluye un indicador de segmento único por subimagen;
- determinar (604), por el decodificador, si el indicador de segmento único por subimagen tiene un primer valor o un segundo valor, donde el primer valor especifica que cada subimagen que se refiere al PPS incluye uno y
- 10 solo un segmento rectangular, y donde el segundo valor especifica que cada subimagen que se refiere al PPS puede incluir uno o más segmentos rectangulares; y
- decodificar (606), mediante el decodificador, el uno y sólo un segmento rectangular cuando el indicador de segmento único por subimagen tiene el primer valor y el uno o más segmentos rectangulares cuando el
- 15 indicador de segmento único por subimagen tiene el segundo valor para obtener una imagen decodificada.
2. El método según la reivindicación 1, en el que el indicador de segmento único por subimagen se designa como `pps_single_slice_per_subpic_flag`.
- 20 3. El método según cualquiera de las reivindicaciones 1 a 2, en el que el primer valor es uno (1) y el segundo valor es cero (0).
4. El método según cualquiera de las reivindicaciones 1 a 3, en el que el PPS comprende además un indicador del número de segmentos en imagen, y en el que un valor del indicador del número de segmentos
- 25 en imagen más uno especifica un número de segmentos en cada imagen que se refiere al PPS.
5. El método según cualquiera de las reivindicaciones 1 a 4, en el que el indicador del número de segmentos en imagen se designa como `pps_num_slices_in_pic_minus1`.
- 30 6. El método según cualquiera de las reivindicaciones 1 a 5, en el que cada subimagen comprende una región rectangular de uno o más segmentos dentro de una imagen.
7. El método según cualquiera de las reivindicaciones 1 a 6, que comprende además generar una imagen para visualización basada en el uno y solo un segmento rectangular cuando el indicador de segmento único
- 35 por subimagen tiene el primer valor y el uno o más segmentos rectangulares cuando el indicador de segmento único por subimagen tiene el segundo valor.
8. Un método implementado por un codificador, comprendiendo el método:
- 40 ajustar, por el codificador, un indicador de segmento único por subimagen a un primer valor cuando cada subimagen que se refiere a un conjunto de parámetros de imagen (PPS) incluye uno y solo un segmento rectangular y a un segundo valor cuando cada subimagen que se refiere al PPS puede incluir uno o más segmentos rectangulares;
- 45 codificar, por el codificador, el PPS que contiene el indicador de segmento único por subimagen en un flujo de bits; y
- almacenar, por el codificador, el flujo de bits para su comunicación hacia un decodificador.
- 50 9. Un dispositivo de decodificación, que comprende:
- un receptor configurado para recibir un flujo de bits que incluye un conjunto de parámetros de imagen, PPS, que contiene un indicador de segmento único por subimagen;
- 55 una memoria acoplada al receptor, almacenando la memoria instrucciones; y
- un procesador acoplado a la memoria, el procesador configurado para ejecutar las instrucciones para hacer que el dispositivo de decodificación:
- 60 determinar por el decodificador, si el indicador de segmento único por subimagen tiene un primer valor o un segundo valor, donde el primer valor especifica que cada subimagen que se refiere al PPS incluye uno y solo un segmento rectangular, y donde el segundo valor especifica que cada subimagen que se refiere al PPS puede incluir uno o más segmentos rectangulares; y

decodificar mediante el decodificador, el uno y sólo un segmento rectangular cuando el indicador de segmento único por subimagen tiene el primer valor y el uno o más segmentos rectangulares cuando el indicador de segmento único por subimagen tiene el segundo valor para obtener una imagen decodificada.

5 10. El dispositivo de decodificación según la reivindicación 9, en el que el primer valor es uno (1) y el segundo valor es cero (0).

10 11. El dispositivo de decodificación, según cualquiera de las reivindicaciones 9 a 10, en el que el PPS comprende además un indicador del número de segmentos en imagen, y en el que un valor del indicador del número de segmentos en imagen más uno especifica un número de segmentos en cada imagen que se refiere al PPS.

15 12. El dispositivo de decodificación según cualquiera de las reivindicaciones 9 a 11, en el que el indicador del número de segmentos en imagen se designa como `pps_num_slices_in_pic_minus1`.

20 13. El dispositivo de decodificación según cualquiera de las reivindicaciones 9 a 12, en el que el procesador está configurado además para generar una imagen para visualización en basa al uno y solo un segmento rectangular cuando el indicador de segmento único por subimagen tiene el primer valor y el uno o más segmentos rectangulares cuando el indicador de segmento único por subimagen tiene el segundo valor.

25 14. Un dispositivo de codificación, que comprende:

una memoria que contiene instrucciones;

25 un procesador acoplado a la memoria, el procesador configurado para implementar las instrucciones para hacer que el dispositivo de codificación:

30 codificar un flujo de bits que incluye un conjunto de parámetros de imagen, PPS, que contiene un indicador de segmento único por subimagen;

ajustar el indicador de segmento único por subimagen a un primer valor cuando cada subimagen que se refiere al PPS incluye uno y solo un segmento rectangular y a un segundo valor cuando cada subimagen que se refiere al PPS puede incluir uno o más segmentos rectangulares; y

35 un transmisor acoplado al procesador, el transmisor configurado para transmitir el flujo de bits hacia un decodificador de vídeo.

15. Un sistema, que comprende:

40 un codificador; y

un decodificador en comunicación con el codificador, en el que el codificador o el decodificador incluye el dispositivo de decodificación, el dispositivo de codificación o el aparato de codificación según cualquiera de las reivindicaciones 9 a 14.

45

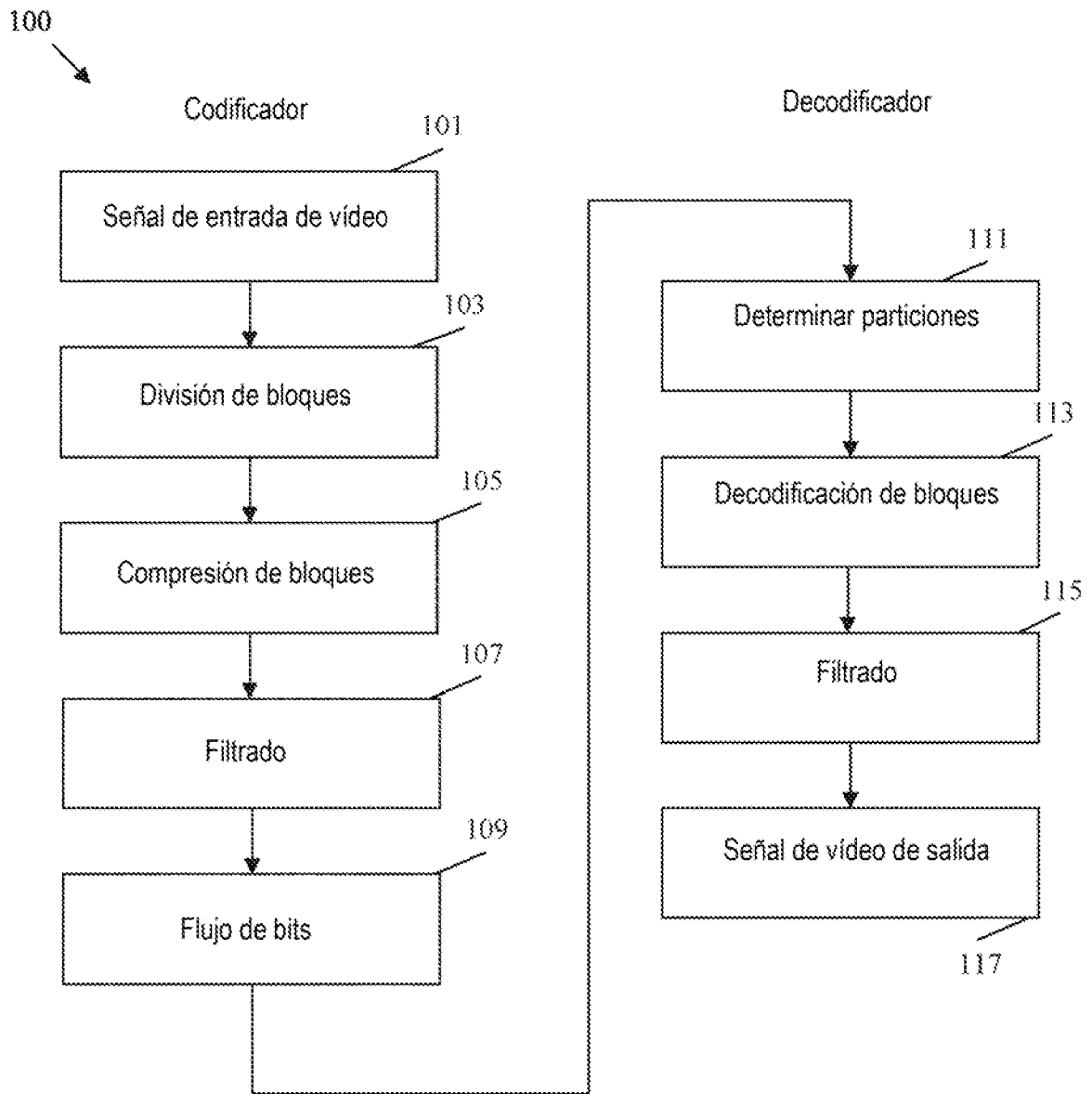


FIG. 1

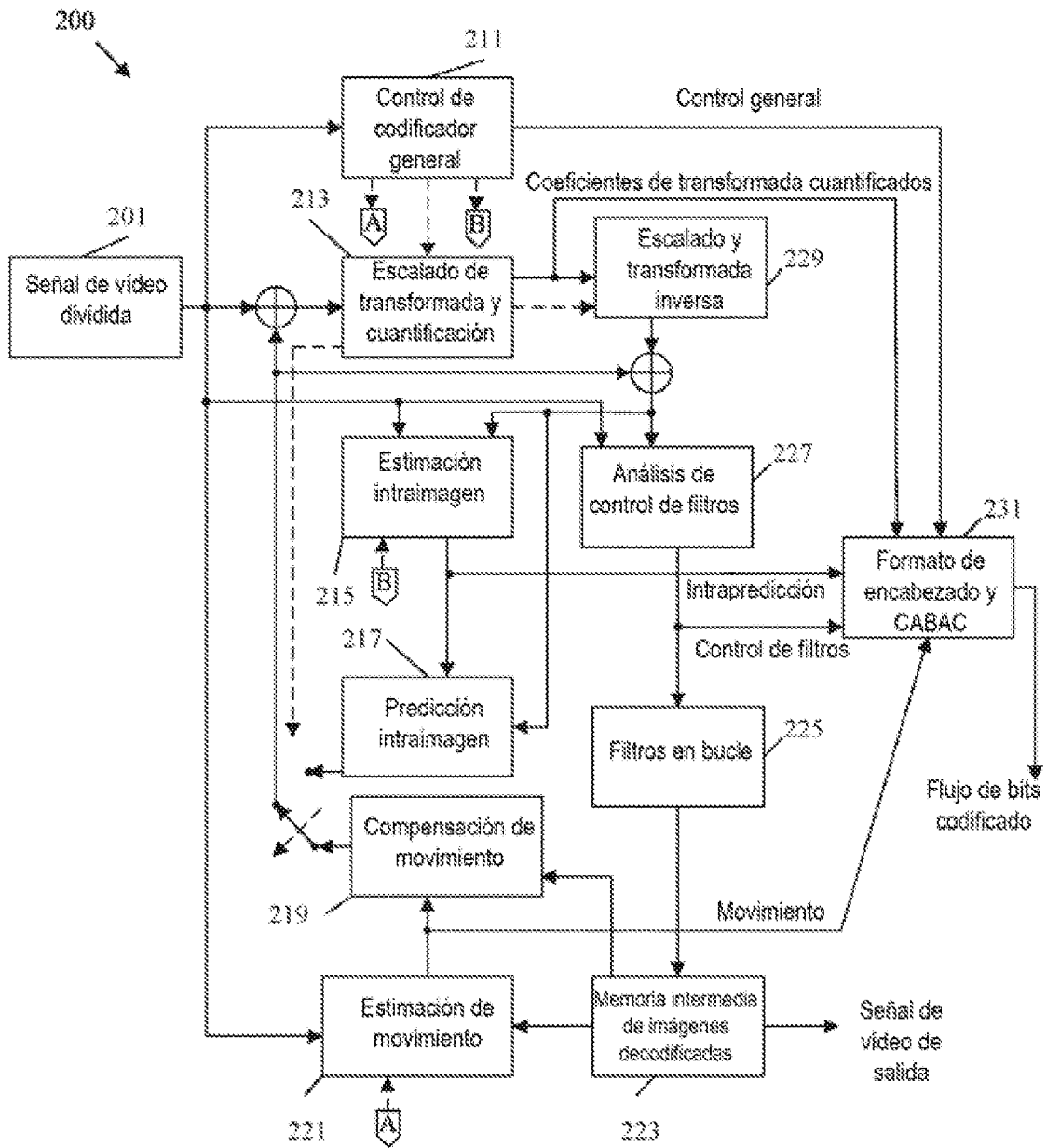


FIG. 2

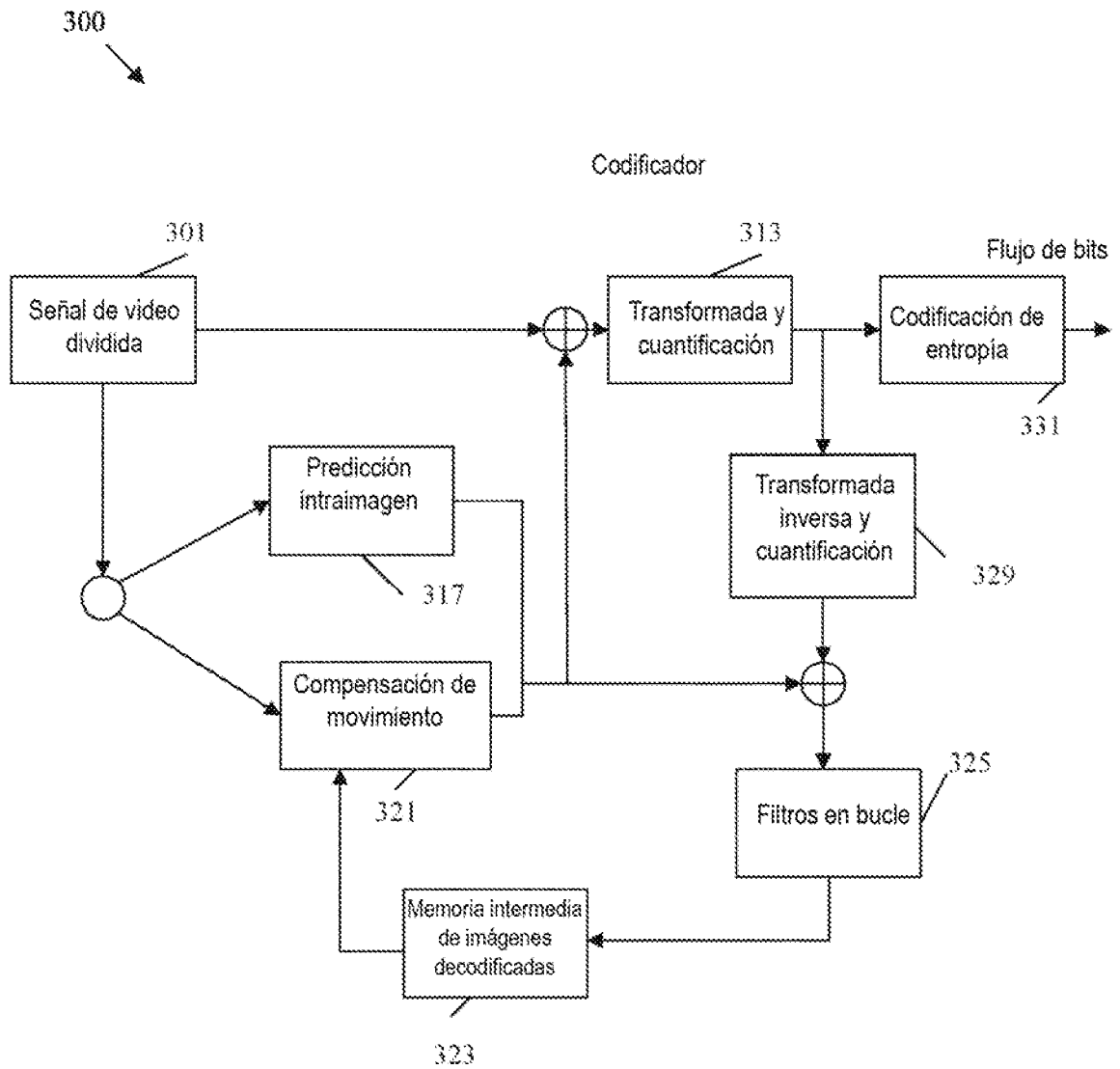


FIG. 3

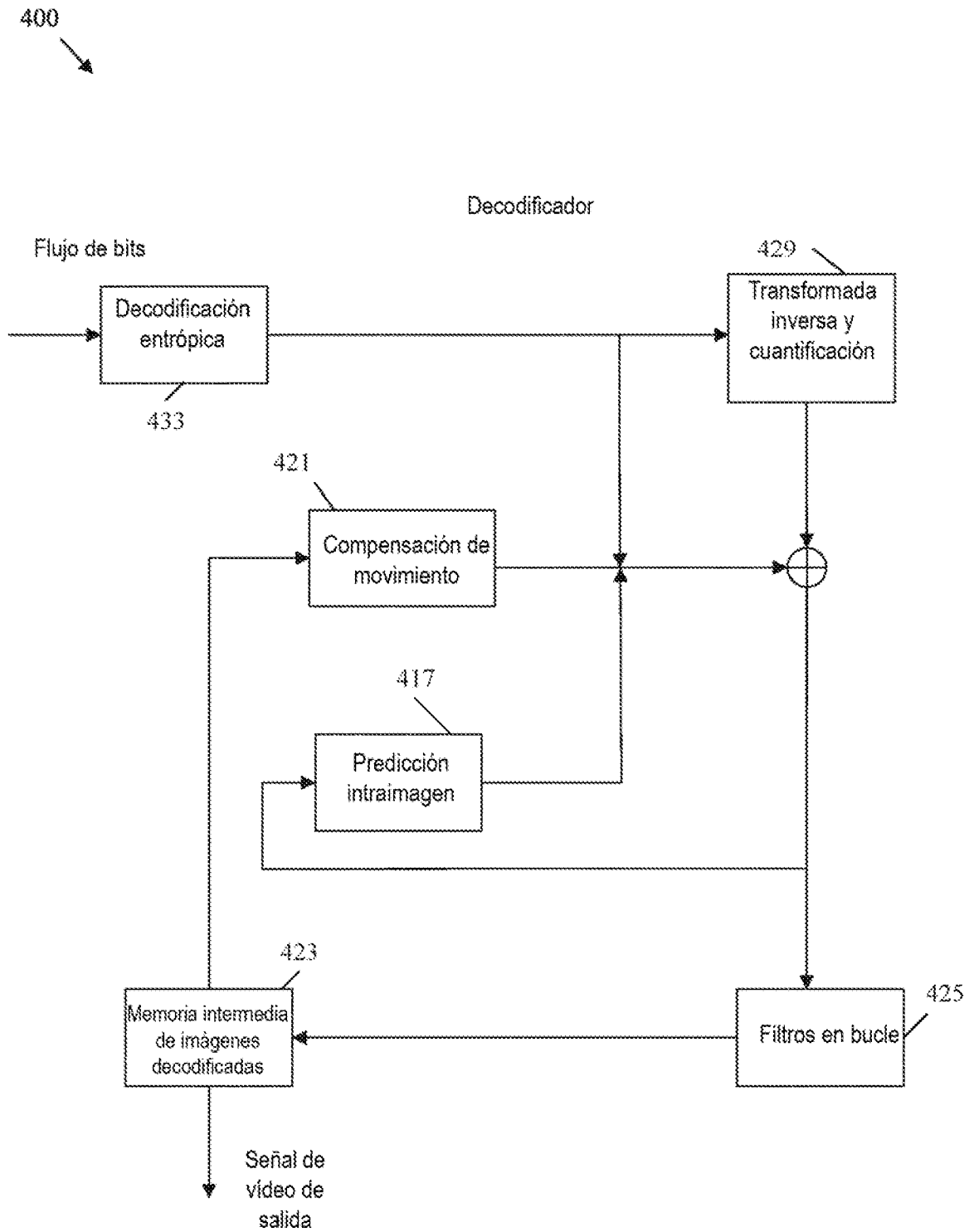


FIG. 4

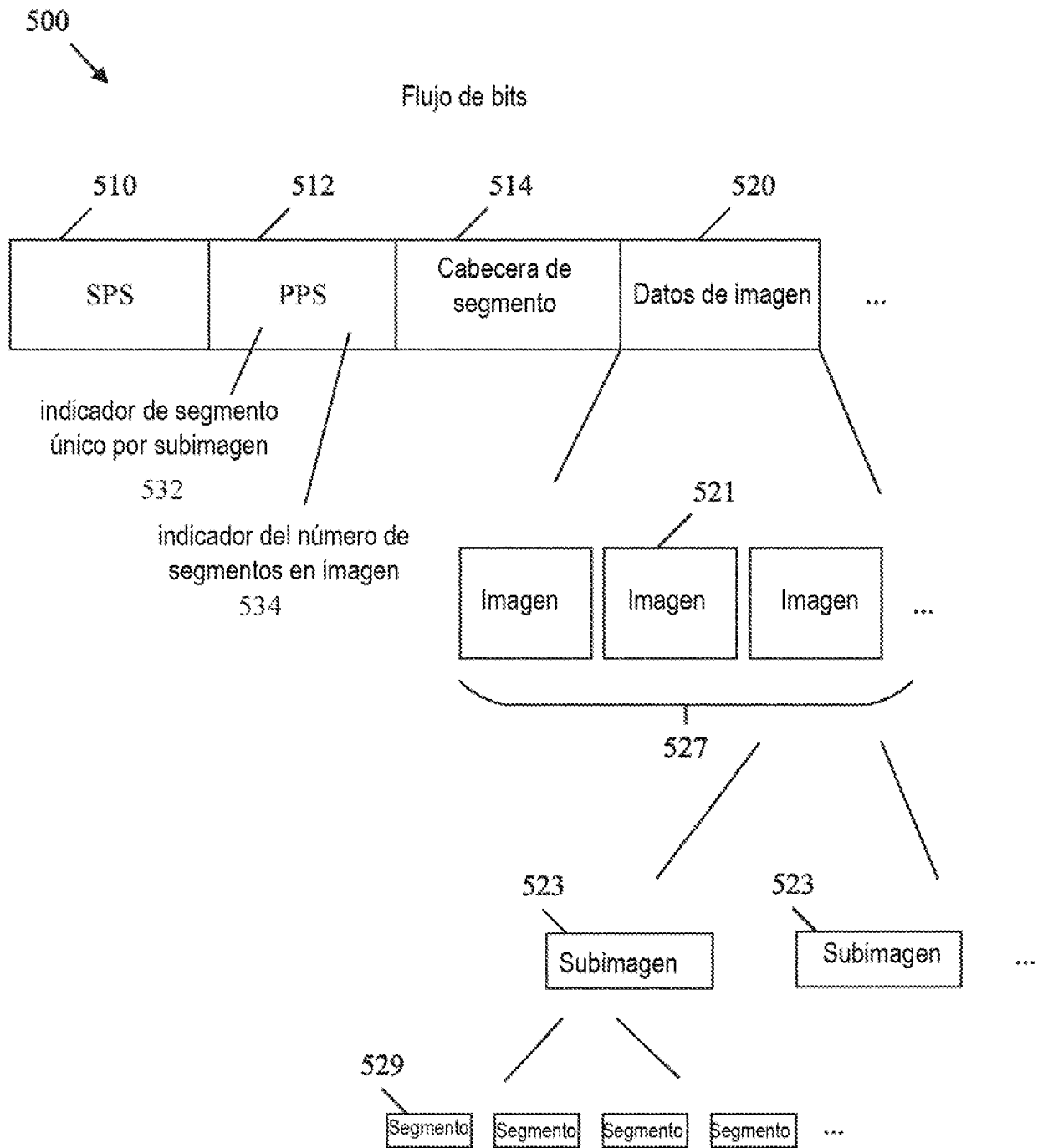


FIG. 5

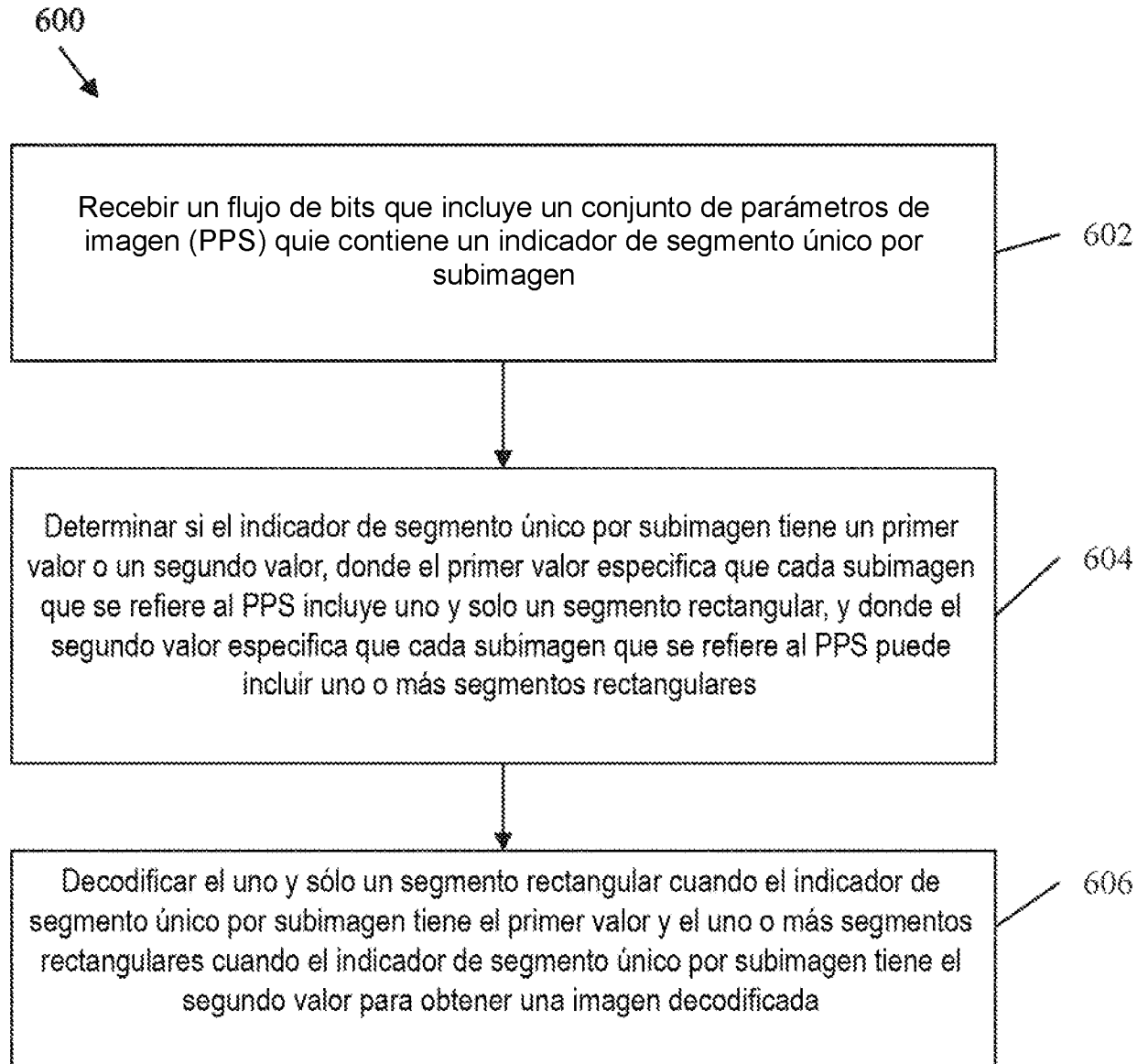


FIG. 6

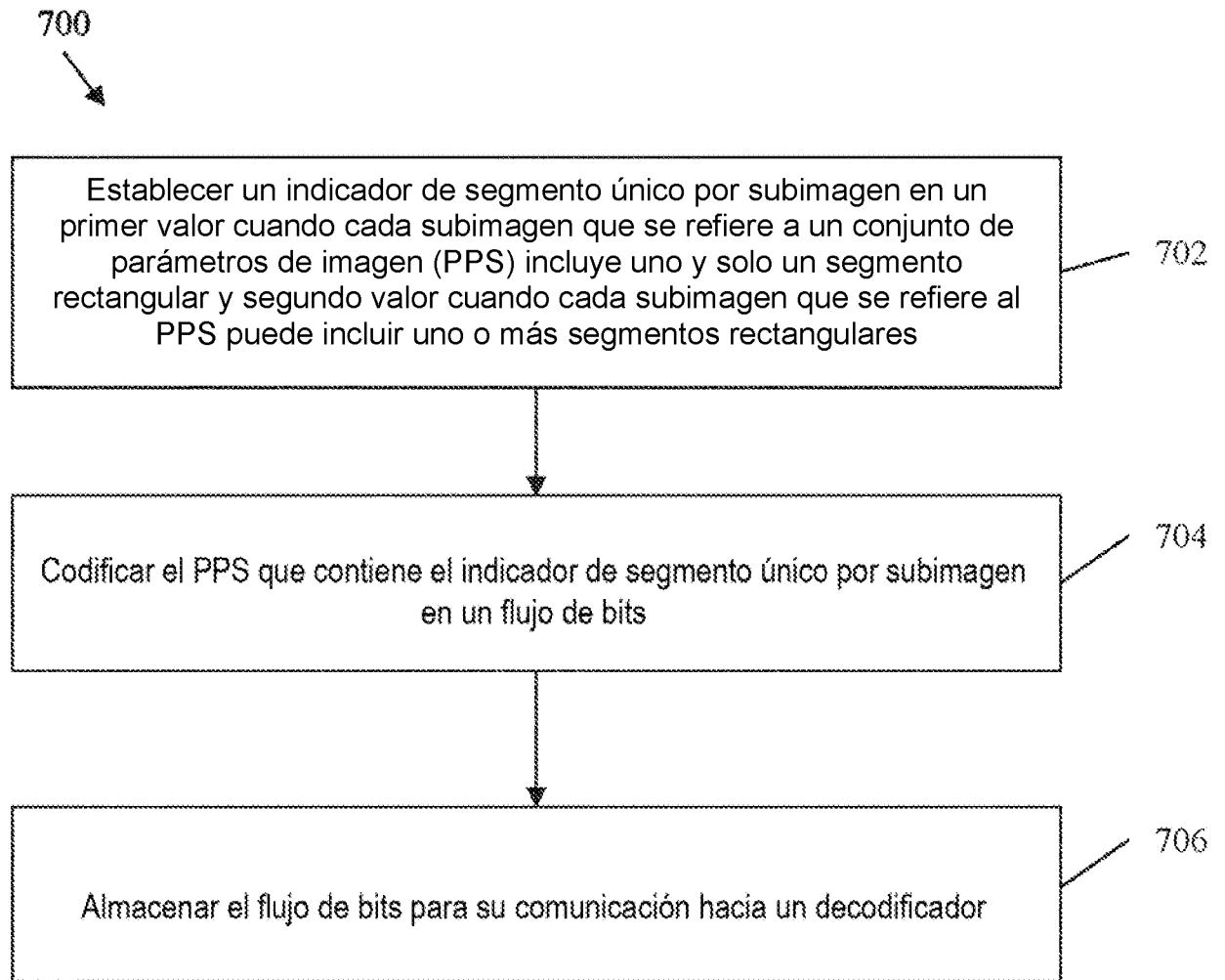


FIG. 7

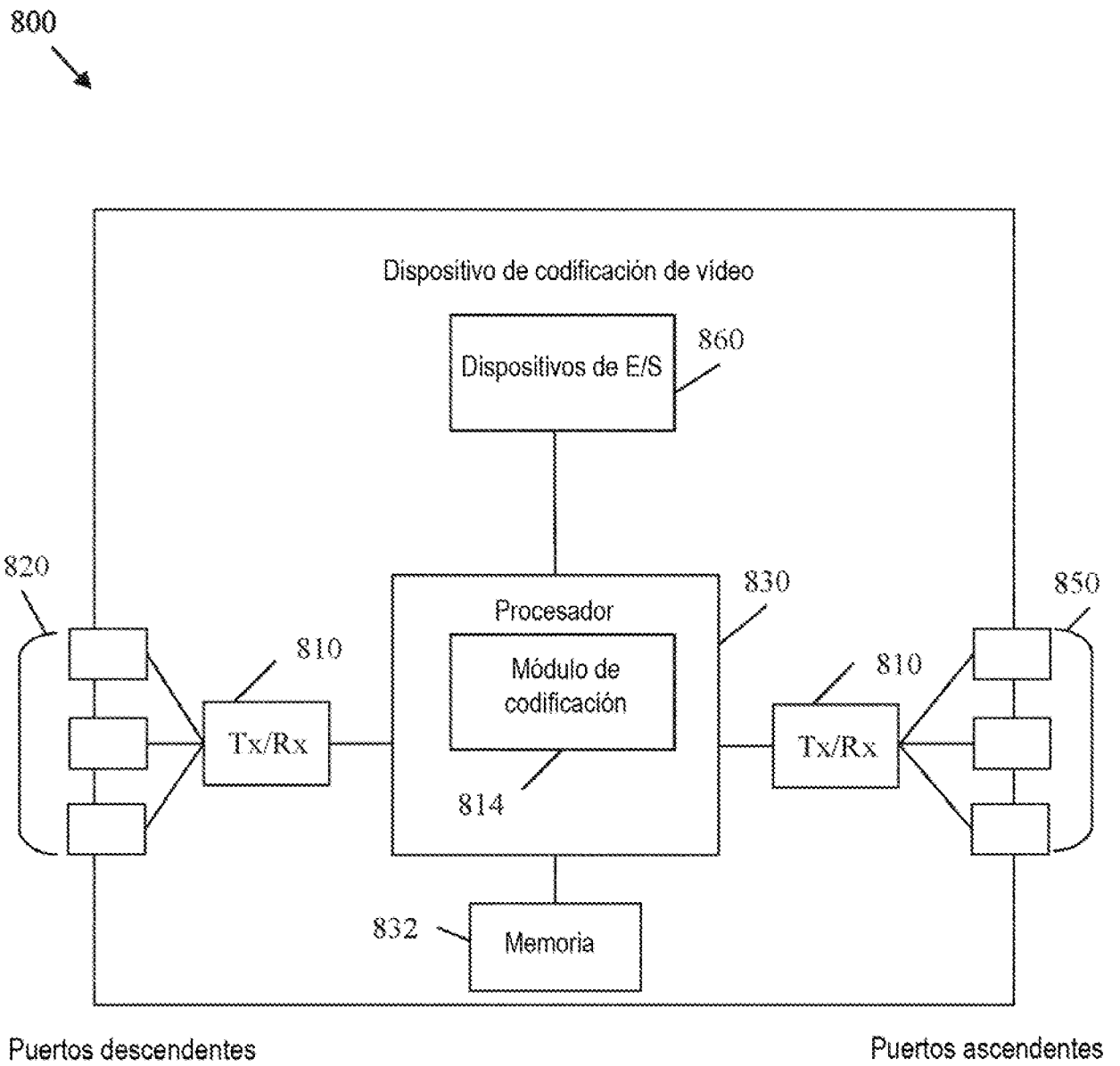


FIG. 8

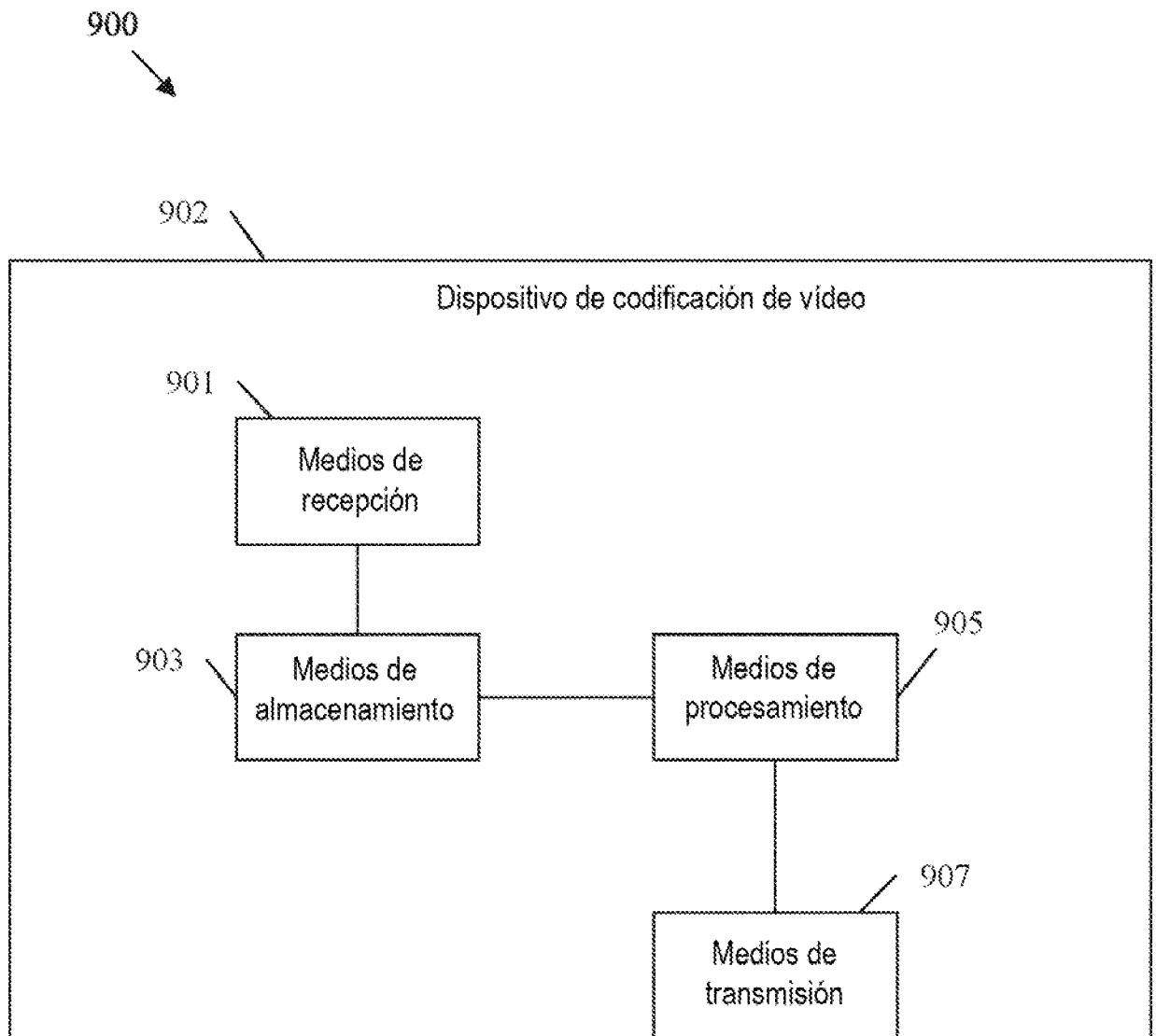


FIG. 9