

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 989 421**

51 Int. Cl.:

H04N 19/11 (2014.01)

H04N 19/13 (2014.01)

H04N 19/70 (2014.01)

H04N 19/159 (2014.01)

H04N 19/593 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **23.03.2020** **PCT/CN2020/080650**

87 Fecha y número de publicación internacional: **01.10.2020** **WO20192625**

96 Fecha de presentación y número de la solicitud europea: **23.03.2020** **E 20779704 (4)**

97 Fecha y número de publicación de la concesión europea: **10.07.2024** **EP 3932057**

54 Título: **Un codificador, un decodificador y métodos correspondientes para la intrapredicción**

30 Prioridad:

23.03.2019 US 201962822865 P

26.03.2019 US 201962824282 P

27.03.2019 US 201962824360 P

45 Fecha de publicación y mención en BOPI de la
traducción de la patente:
26.11.2024

73 Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building, Bantian,
Longgang District, Shenzhen
Guangdong 518129, CN

72 Inventor/es:

WANG, BIAO;
ESENLIK, SEMIH;
KOTRA, ANAND MEHER;
GAO, HAN y
CHEN, JIANLE

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 989 421 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

Un codificador, un decodificador y métodos correspondientes para la intrapredicción

5 **Campo técnico de la invención**

Las realizaciones de la presente solicitud (divulgación) se refieren generalmente al campo del procesamiento de imágenes y más particularmente a la intrapredicción.

10 **Antecedentes de la invención**

La codificación de video (codificación y decodificación de video) se utiliza en una amplia gama de aplicaciones de video digital, por ejemplo, transmisión de TV digital, transmisión de video a través de Internet y redes móviles, aplicaciones de conversación en tiempo real como video chat, videoconferencia, DISCOS DVD y Blu-ray, sistemas de edición y adquisición de contenido de video y videocámaras de aplicaciones de seguridad.

La cantidad de datos de video necesarios para representar incluso un video relativamente corto puede ser sustancial, lo que puede resultar en dificultades cuando los datos deben transmitirse o comunicarse a través de una red de comunicaciones con capacidad de ancho de banda limitada. Por lo tanto, los datos de video generalmente se comprimen antes de comunicarse a través de las redes de telecomunicaciones de hoy en día. El tamaño de un video también podría ser un problema cuando el video se almacena en un dispositivo de almacenamiento porque los recursos de memoria pueden ser limitados. Los dispositivos de compresión de video a menudo usan software y/o hardware en la fuente/origen para codificar los datos de video antes de su transmisión o almacenamiento, disminuyendo así la cantidad de datos necesarios para representar imágenes de video digitales. Los datos comprimidos son luego recibidos en el destino por un dispositivo de descompresión de video que decodifica los datos de video. Con recursos de red limitados y demandas cada vez mayores de mayor calidad de video, son deseables técnicas mejoradas de compresión y descompresión que mejoren la relación de compresión con poco o ningún sacrificio en la calidad de las imágenes.

En particular, existe una demanda continua para reducir la carga del procesador en el contexto del procesamiento de intrapredicción. En la técnica, se conoce el empleo de la lista del modo más probable (MPM) para la codificación del modo de intrapredicción. La lista MPM reduce los bits necesarios para codificar el modo de intrapredicción del bloque actual. Cuando el modo de intrapredicción del bloque actual corresponde a una entrada de una lista MPM, se codifica un índice en lugar del modo real, por lo que se necesitan menos bits. Sin embargo, la selección de las entradas para la generación de una lista MPM en vista de la codificación de intrapredicción más eficiente es todavía una tarea. La selección adecuada de las entradas influye en gran medida en la eficacia de la codificación del modo de intrapredicción basada en la lista MPM.

En vista de esto, es un objeto subyacente de la presente invención proporcionar una técnica para obtener un modo de intrapredicción basado en una lista MPM que dé como resultado una codificación más eficiente en comparación con la técnica actual.

El documento de WANG (HUAWEI) B ET AL.: "*CE3-related: A unified MPM list for intra mode coding*", 14 REUNIÓN DE JVET; 20190319 – 20190327; GINEBRA; n.º JVET-N0185, 12 de marzo de 2019, XP030202669, proporciona un texto preliminar de una especificación de codificación de video.

El documento de YU (MOTOROLA) Y ET AL.: "*On MPM determination and Planar mode signaling*" 8 REUNIÓN DE JCT-VC; 20120201 – 20120201; SAN JOSE; n.º JCTVC-H0516, 3 de febrero de 2012 (03-02-2012), XP030232397, describe los detalles de implementación y da a conocer los resultados de una determinación MPM y una señalización de modo planar.

Breve descripción de la invención

La invención se expone en el conjunto de reivindicaciones adjuntas. Las realizaciones de la presente solicitud proporcionan dispositivos y métodos para codificar y decodificar según las reivindicaciones independientes.

Los objetos anteriores y otros se consiguen mediante el objeto de las reivindicaciones independientes. Otras formas de implementación son evidentes a partir de las reivindicaciones dependientes, la descripción y las figuras.

Según un primer aspecto, la invención se refiere a un método realizado por un dispositivo de decodificación. El método incluye: obtener un valor de una primera información de indicación de un bloque actual, el valor de la primera información de indicación que indica si un modo de intrapredicción del bloque actual está comprendido en un conjunto de modos más probables; y obtener un valor de una línea de índice de referencia del bloque actual. El conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y el modo planar. Cuando el valor de la primera información de indicación indica que el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y cuando el valor de la línea de índice de referencia indica que se utiliza la línea

de referencia vecina más cercana al bloque actual, el método incluye además obtener un valor de una segunda información de indicación del bloque actual, indicando el valor de la segunda información de indicación del bloque actual si el modo de intrapredicción del bloque actual es el modo planar o no.

En el método convencional, la complejidad de decodificación para el procesamiento MPM es alta. Dependiendo del uso de las herramientas de línea de referencia múltiple (MRL) y del modo de codificación intra subpartición (ISP), se construyen tres listas MPM diferentes (6 entradas). En los aspectos o posibles realizaciones de los aspectos de esta invención, la construcción de la lista MPM (5 entradas) se unifica con una complejidad de decodificación reducida, independientemente de si se usa la herramienta de codificación MRL o ISP.

Además, el modo planar se considera un modo especial debido a su alta frecuencia y se asigna una bandera especial para comprobar si el modo actual es planar o no. Aunque no está en la lista MPM (con 5 entradas inventadas), siempre se marca cuando la bandera mpm es verdadera. En otras palabras, la bandera mpm indica si el modo de intrapredicción del bloque actual está en una lista de 6 entradas, que comprende el modo planar y las 5 entradas de la lista MPM. Las 5 entradas de la lista MPM se construyen en función de los modos de intrapredicción del bloque izquierdo y superior.

Como implementación del primer aspecto, la primera información de indicación se indica mediante una primera bandera `intra_luma_mpm_flag`. El valor de la línea de índice de referencia del bloque actual puede indicarse mediante `intra_luma_ref_idx`. La segunda información de indicación se indica mediante una segunda bandera `intra_luma_not_planar_flag`.

Cuando la codificación MRL no está habilitada, es decir, cuando el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual, los dispositivos analizan un flujo de bits para obtener el valor de la segunda información de indicación. Cuando la codificación MRL está habilitada, el valor de la segunda información de indicación se obtiene mediante un valor predeterminado. En esta situación, se excluye el modo planar y su bandera correspondiente (es decir, `intra_luma_not_planar_flag`) no se analiza a partir del flujo de bits, pero se infiere como un valor predeterminado (por ejemplo, un valor de 1). En este caso, se guarda un bit para indicar el valor de la bandera planar (es decir, `intra_luma_not_planar`) cuando MRL está habilitado.

Según un segundo aspecto, la invención se refiere a un método realizado por un aparato de codificación. El método incluye: determinar si un modo de intrapredicción de un bloque actual está comprendido en un conjunto de modos más probables; y determinar si la línea de referencia vecina más cercana al bloque actual se usa para la intrapredicción. El conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y el modo planar. El método incluye además codificar un flujo de bits, en donde el flujo de bits incluye información para indicar si el modo de intrapredicción del bloque actual es el modo planar cuando el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y la línea de referencia vecina más cercana al bloque actual se utiliza para la intrapredicción.

Como una implantación del segundo aspecto, el flujo de bits incluye una primera bandera que indica si el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y una segunda bandera que indica si el modo de intrapredicción del bloque actual es el modo planar. La primera bandera es `intra_luma_mpm_flag` y la segunda bandera es `intra_luma_not_planar_flag`.

El método según el primer aspecto de la invención puede ser realizado por un dispositivo de decodificación según el tercer aspecto de la invención. El dispositivo de decodificación incluye una primera unidad de obtención y una segunda unidad de obtención. La primera unidad de obtención, configurada para obtener un valor de una línea de índice de referencia de un bloque actual, y un valor de una primera información de indicación del bloque actual, el valor de la primera información de indicación indica si un modo de intrapredicción del bloque actual está comprendido en un conjunto de modos más probables. El conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y el modo planar. La segunda unidad de obtención, configurada para obtener un valor de una segunda información de indicación del bloque actual cuando el valor de la primera información de indicación indica que el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y cuando el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual, en donde el valor de la segunda información de indicación del bloque actual indica si el modo de intrapredicción del bloque actual es el modo planar.

Otras características y formas de implementación del método según el tercer aspecto de la invención corresponden a las características y formas de implementación del aparato según el primer aspecto de la invención.

El método según el segundo aspecto de la invención puede ser realizado por un dispositivo de codificación según el cuarto aspecto de la invención. El dispositivo de codificación incluye una unidad de determinación y una unidad de codificación. La unidad de determinación, configurada para determinar si un modo de intrapredicción de un bloque actual está comprendido en un conjunto de modos más probables, y para determinar si la línea de referencia vecina más cercana al bloque actual se usa para la intrapredicción. El conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y el modo planar. La unidad de codificación, configurada para codificar un flujo

de bits, en donde el flujo de bits incluye información para indicar si el modo de intrapredicción del bloque actual es el modo planar cuando el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y la línea de referencia vecina más cercana al bloque actual se utiliza para la intrapredicción.

5 Otras características y formas de implementación del método según el cuarto aspecto de la invención corresponden a las características y formas de implementación del aparato según el segundo aspecto de la invención.

10 Según un quinto aspecto, la invención se refiere a un programa informático/de ordenador que comprende un código de programa para realizar el método según el primer o segundo aspecto o cualquier posible realización del primer o segundo aspecto cuando se ejecuta en un ordenador.

15 El modo planar tiene una alta frecuencia que se utiliza como modo de intrapredicción. La eficiencia de codificación del proceso de codificación se puede mejorar mediante el empleo de la información para indicar si el modo de intrapredicción del bloque actual es el modo planar en comparación con la técnica actual, cuando el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y la línea de referencia vecina más cercana al bloque actual se utiliza para la intrapredicción.

20 Los detalles de una o más realizaciones se establecen en los dibujos adjuntos y la descripción a continuación. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

20 Breve descripción de los dibujos

25 En lo siguiente, realizaciones de la invención se describen con más detalle con referencia a las figuras y dibujos adjuntos, en los que:

La FIG. 1A es un diagrama de bloques que muestra un ejemplo de un sistema de codificación de video configurado para implementar realizaciones de la invención;

30 La FIG. 1B es un diagrama de bloques que muestra otro ejemplo de un sistema de codificación de video configurado para implementar realizaciones de la invención;

La FIG. 2 es un diagrama de bloques que muestra un ejemplo de un codificador de video para implementar realizaciones de la invención;

35 La FIG. 3 es un diagrama de bloques que muestra un ejemplo de un decodificador de video configurado para implementar realizaciones de la invención;

40 La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de un aparato de codificación o un aparato de decodificación;

La FIG. 5 es un diagrama de bloques que ilustra otro ejemplo de un aparato de codificación o un aparato de decodificación;

45 La FIG. 6 es un diagrama de bloques que ilustra un ejemplo de herramienta de codificación de línea de referencia múltiple (MRL);

La FIG. 7 es un diagrama de bloques que ilustra un ejemplo de división de bloques 4×8 y 8×4 ;

50 La FIG. 8 es un diagrama de bloques que ilustra un ejemplo de división de todos los bloques excepto 4×8 , 8×4 y 4×4 ;

La FIG. 9 es un diagrama de bloques que ilustra un ejemplo de modelado/codificación de contexto CABAC;

55 La FIG. 10 es un diagrama de bloques que ilustra un ejemplo de los bloques vecinos izquierdo (L) y superior (A) del bloque actual;

La FIG. 11 es un diagrama de bloques que ilustra los modos de intrapredicción;

60 La FIG. 12A ilustra realizaciones de un método 1200 de intrapredicción de un bloque actual implementado por un dispositivo de decodificación según la presente invención;

La FIG. 12B ilustra realizaciones de un método 1210 de intrapredicción de un bloque actual implementado por un dispositivo de decodificación según la presente invención;

65 La FIG. 13 ilustra realizaciones de un método 1300 de intrapredicción de un bloque actual implementado por un dispositivo de codificación según la presente invención;

La FIG. 14 ilustra realizaciones de un dispositivo de decodificación 1400 para su uso en un decodificador de imágenes según la presente invención;

5 La FIG. 15 ilustra realizaciones de un dispositivo de codificación 1500 para su uso en un codificador de imágenes según la presente invención;

La FIG. 16 es un diagrama de bloques que muestra una estructura de ejemplo de un sistema de suministro de contenido 1300 que realiza un servicio de entrega de contenido; y

10 La FIG. 17 es un diagrama de bloques que muestra una estructura de un ejemplo de un dispositivo terminal.

En lo que sigue, los signos de referencia idénticos se refieren a características idénticas o al menos funcionalmente equivalentes, si no se especifica explícitamente lo contrario.

15 Descripción detallada de las realizaciones

En la siguiente descripción, se hace referencia a los dibujos adjuntos que forman parte de esta divulgación, y que muestran, a modo de ilustración, aspectos específicos de realizaciones de la invención o aspectos específicos en los que se pueden utilizar las realizaciones de la presente invención. Debe entenderse que las realizaciones de la
20 invención pueden usarse en otros aspectos y comprender cambios estructurales o lógicos no representados en los figuras. Por lo tanto, la siguiente descripción detallada no debe ser tomada en un sentido de limitación, y el alcance de la presente invención está definido por las reivindicaciones adjuntas.

Por ejemplo, se entiende que una divulgación en relación con un método descrito también puede ser válida para un
25 dispositivo o sistema correspondiente configurado para realizar el método y viceversa. Por ejemplo, si se describen uno o una pluralidad de pasos específicos del método, un dispositivo correspondiente puede incluir una o una pluralidad de unidades, por ejemplo, unidades funcionales, para realizar uno o una pluralidad de pasos del método descritos (por ejemplo, una unidad que realice uno o una pluralidad de pasos, o una pluralidad de unidades que
30 realicen cada una uno o más de la pluralidad de pasos), incluso si dichas unidades no se describen o se ilustran explícitamente en las figuras. Por otra parte, por ejemplo, si un aparato específico se describe con base en una o una pluralidad de unidades, por ejemplo, unidades funcionales, un método correspondiente puede incluir un paso para realizar la funcionalidad de la una o la pluralidad de unidades (por ejemplo, un paso que realiza la funcionalidad de la una o la pluralidad de unidades, o una pluralidad de pasos que realizan cada uno la funcionalidad de una o más de la pluralidad de unidades), incluso si tal uno o pluralidad de pasos no se describen o se ilustran explícitamente en las
35 figuras. Además, se entiende que las características de las diversas realizaciones y/o aspectos ejemplares descritos en el presente documento pueden combinarse entre sí, a menos que se indique específicamente lo contrario.

La codificación de video suele referirse al procesamiento de una secuencia de imágenes, que forman el video o la secuencia de video. En lugar del término "foto/fotografía", el término "fotograma/cuadro" o "imagen" se puede utilizar
40 como sinónimos en el campo de la codificación de video. La codificación de video (o codificación en general) comprende dos partes de codificación/encodificación y decodificación de video. La codificación de video se realiza en el lado de la fuente, y suele comprender el procesamiento (por ejemplo, mediante compresión) de las imágenes de video originales para reducir la cantidad de datos necesarios para representar las imágenes de video (para un almacenamiento y/o transmisión más eficientes). La decodificación de video se realiza en el lado de destino y suele
45 comprender el procesamiento inverso al del codificador para reconstruir las imágenes de video. Las realizaciones que se refieren a la "codificación" de imágenes de video (o imágenes en general) se entenderán relacionadas con la "codificación" o la "decodificación" de imágenes de video o secuencias de video respectivas. La combinación de la parte de codificación y la parte de decodificación también se denomina CODEC (Codificación y Decodificación)

50 En el caso de la codificación de video sin pérdidas, las imágenes de video originales pueden reconstruirse, es decir, las imágenes de video reconstruidas tienen la misma calidad que las imágenes de video originales (suponiendo que no haya pérdidas de transmisión u otras pérdidas de datos durante el almacenamiento o la transmisión). En caso de codificación de video con pérdidas, compresión adicional, por ejemplo, mediante cuantificación/cuantización, para reducir la cantidad de datos que representan las imágenes de video, que no pueden reconstruirse por completo en el
55 decodificador, es decir, la calidad de las imágenes de video reconstruidas es menor o peor en comparación con la calidad de las imágenes de video originales.

Varios estándares de codificación de video pertenecen al grupo de los "códecs de video híbridos con pérdidas" (es decir, combinan la predicción espacial y temporal en el dominio de la muestra y la codificación por transformación 2D
60 para aplicar la cuantificación en el dominio de la transformación). Cada imagen de una secuencia de video típicamente se particiona/divide en un conjunto de bloques no superpuestos y la codificación suele realizarse a nivel de bloque. En otras palabras, en el codificador, el video se procesa típicamente, es decir, se codifica, en un nivel de bloque (bloque de video), por ejemplo, mediante el uso de la predicción espacial (intraimagen) y/o la predicción temporal (interimagen) para generar un bloque de predicción, restando el bloque de predicción del bloque actual (bloque actualmente procesado/por procesar) para obtener un bloque residual, transformando el bloque residual y
65 cuantificando el bloque residual en el dominio de transformación para reducir la cantidad de datos a transmitir

(compresión), mientras que en el decodificador el procesamiento inverso comparado con el codificador se aplica al bloque codificado o comprimido para reconstruir el bloque actual para su representación. Además, el codificador duplica el bucle/lazo de procesamiento del decodificador de modo que ambos generarán predicciones idénticas (por ejemplo, intra e inter predicciones) y/o reconstrucciones para procesar, es decir, codificar los bloques subsiguientes.

En las siguientes realizaciones de un sistema de codificación de video 10, se describen un codificador de video 20 y un decodificador de video 30 basándose en las FIGS. 1 a 3.

La FIG. 1A es un diagrama de bloques esquemático que ilustra un ejemplo de sistema de codificación 10, por ejemplo, un sistema de codificación de video 10 (o sistema de codificación corta 10) que puede utilizar técnicas de esta presente solicitud. El codificador de video 20 (o codificador corto 20) y el decodificador de video 30 (o decodificador corto 30) del sistema de codificación de video 10 representan ejemplos de dispositivos que pueden configurarse para realizar técnicas de acuerdo con varios ejemplos descritos en la presente solicitud.

Como se muestra en la FIG. 1A, el sistema de codificación 10 comprende un dispositivo de origen 12 configurado para proporcionar datos de imagen codificados 21, por ejemplo, a un dispositivo de destino 14 para decodificar los datos de imagen codificados 13.

El dispositivo de origen 12 comprende un codificador 20, y puede comprender además, es decir, opcionalmente, una fuente de imágenes 16, un preprocesador (o unidad de preprocesamiento) 18, por ejemplo, un preprocesador de imágenes 18, y una interfaz de comunicación o unidad de comunicación 22.

La fuente de imágenes 16 puede comprender o ser cualquier tipo de dispositivo de captura de imágenes, por ejemplo, una cámara para capturar una imagen del mundo real, y/o cualquier tipo de dispositivo de generación de imágenes, por ejemplo, un procesador de gráficos por ordenador para generar una imagen animada por ordenador, o cualquier otro dispositivo para obtener y/o proporcionar una imagen del mundo real, una imagen generada por ordenador (por ejemplo, un contenido de pantalla, una imagen de realidad virtual (VR)) y/o cualquier combinación de las mismas (por ejemplo, una imagen de realidad aumentada (AR)). La fuente de imágenes puede ser cualquier tipo de memoria o almacenamiento que almacene cualquiera de las imágenes mencionadas anteriormente.

A diferencia del preprocesador 18 y del procesamiento realizado por la unidad de preprocesamiento 18, la imagen o los datos de la imagen 17 también pueden denominarse imagen sin procesar o datos de la imagen sin procesar 17.

El preprocesador 18 está configurado para recibir los datos de la imagen (sin procesar) 17 y realizar un preprocesamiento de los datos de imagen 17 para obtener una imagen preprocesada 19 o datos de imagen preprocesados 19. El preprocesamiento realizado por el preprocesador 18 puede comprender, por ejemplo, recorte, conversión de formato de color (por ejemplo, de RGB a YCbCr), corrección de color o eliminación de ruido. Puede entenderse que la unidad de preprocesamiento 18 puede ser un componente opcional.

El codificador de video 20 está configurado para recibir los datos de imagen preprocesados 19 y proporcionar datos de imagen codificados 21 (se describirán más detalles a continuación, por ejemplo, en base a la FIG. 2).

La interfaz de comunicación 22 del dispositivo de origen 12 puede configurarse para recibir los datos de imagen codificados 21 y para transmitir los datos de imagen codificados 21 (o cualquier versión procesada adicional de los mismos) a través del canal de comunicación 13 a otro dispositivo, por ejemplo, el dispositivo de destino 14 o cualquier otro dispositivo, para almacenamiento o reconstrucción directa.

El dispositivo de destino 14 comprende un decodificador 30 (por ejemplo, un decodificador de video 30), y puede comprender además, es decir, opcionalmente, una interfaz o unidad de comunicación 28, un postprocesador 32 (o unidad de postprocesamiento 32) y un dispositivo de visualización 34.

La interfaz de comunicación 28 del dispositivo de destino 14 está configurada para recibir los datos de imagen codificados 21 (o cualquier versión procesada adicional de los mismos), por ejemplo, directamente desde el dispositivo de origen 12 o desde cualquier otra fuente, por ejemplo, un dispositivo de almacenamiento, por ejemplo, un dispositivo de almacenamiento de datos de imágenes codificadas, y proporcionar los datos 21 de imágenes codificados al decodificador 30.

La interfaz de comunicación 22 y la interfaz de comunicación 28 pueden configurarse para transmitir o recibir los datos de imagen codificados 21 o los datos codificados 13 a través de un enlace de comunicación directo entre el dispositivo de origen 12 y el dispositivo de destino 14, por ejemplo, una conexión directa por cable o inalámbrica, o mediante cualquier tipo de red, por ejemplo, una red cableada o inalámbrica o cualquier combinación de las mismas, o cualquier tipo de red pública y privada, o cualquier tipo de combinación de las mismas.

La interfaz de comunicación 22 puede configurarse, por ejemplo, para empaquetar los datos de imagen codificados 21 en un formato apropiado, por ejemplo, paquetes y/o procesar los datos de imagen codificados utilizando cualquier

tipo de codificación de transmisión o procesamiento para su transmisión a través de un enlace de comunicación o red de comunicación.

La interfaz de comunicación 28, que forma la contraparte de la interfaz de comunicación 22, puede configurarse, por ejemplo, para recibir los datos transmitidos y procesar los datos de transmisión utilizando cualquier tipo de decodificación de transmisión correspondiente o procesamiento y/o desempaquetamiento para obtener los datos de imagen codificada 21.

Tanto la interfaz de comunicación 22 como la interfaz de comunicación 28 pueden configurarse como interfaces de comunicación unidireccionales como lo indica la flecha para el canal de comunicación 13 en la FIG. 1A que apunta desde el dispositivo de origen 12 al dispositivo de destino 14, o interfaces de comunicación bidireccionales, y se puede configurar, por ejemplo, para enviar y recibir mensajes, por ejemplo, para establecer una conexión, para reconocer e intercambiar cualquier otra información relacionada con el enlace de comunicación y/o la transmisión de datos, por ejemplo, transmisión de datos de imágenes codificadas.

El decodificador 30 está configurado para recibir los datos de imagen codificados 21 y proporcionar datos de imagen decodificada 31 o una imagen decodificada 31 (más adelante se describirán más detalles, por ejemplo, basándose en la FIG. 3 o la FIG. 5).

El post-procesador 32 del dispositivo de destino 14 está configurado para post-procesar los datos de imagen decodificada 31 (también llamados datos de la imagen reconstruida), por ejemplo la imagen decodificada 31, para obtener los datos de la imagen post-procesada 33, por ejemplo una imagen post-procesada 33. El postprocesamiento realizado por la unidad de postprocesamiento 32 puede comprender, por ejemplo, la conversión del formato de color (por ejemplo, de YCbCr a RGB), la corrección del color, el recorte o el remuestreo, o cualquier otro procesamiento, por ejemplo, para preparar los datos de imagen decodificada 31 para su visualización, por ejemplo, por el dispositivo de visualización 34.

El dispositivo de visualización 34 del dispositivo de destino 14 está configurado para recibir los datos de imagen post-procesada 33 para mostrar la imagen, por ejemplo, a un usuario o espectador. El dispositivo de visualización 34 puede ser o comprender cualquier tipo de pantalla para representar la imagen reconstruida, por ejemplo, una pantalla o monitor integrado o externo. Las pantallas pueden, por ejemplo, comprender pantallas de cristal líquido (LCD), pantallas de diodos emisores de luz orgánicos (OLED), pantallas de plasma, proyectores, pantallas micro LED, cristal líquido sobre silicio (LCoS), procesador de luz digital (DLP) o cualquier otro tipo de pantalla.

Aunque la FIG. 1A representa el dispositivo de origen 12 y el dispositivo de destino 14 como dispositivos separados, las realizaciones de los dispositivos también pueden comprender ambas o ambas funcionalidades, el dispositivo de origen 12 o la funcionalidad correspondiente y el dispositivo de destino 14 o la funcionalidad correspondiente. En tales realizaciones, el dispositivo de origen 12 o la funcionalidad correspondiente y el dispositivo de destino 14 o la funcionalidad correspondiente pueden implementarse utilizando el mismo hardware y/o software o mediante hardware y/o software separados o cualquier combinación de los mismos.

Como resultará evidente para el experto basándose en la descripción, la existencia y división (exacta) de funcionalidades de las diferentes unidades o funcionalidades dentro del dispositivo de origen 12 y/o dispositivo de destino 14 como se muestra en la FIG. 1A puede variar dependiendo de la dispositivo y aplicación reales.

El codificador 20 (por ejemplo, un codificador de video 20) o el decodificador 30 (por ejemplo, un decodificador de video 30) o tanto el codificador 20 como el decodificador 30 pueden implementarse mediante circuitos de procesamiento como se muestra en la FIG. 1B, como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), arreglos de compuertas programables en campo (FPGA), lógica discreta, hardware, codificación de video dedicada o cualquier combinación de los mismos. El codificador 20 puede implementarse a través de circuitería/circuito de procesamiento 46 para incorporar los diversos módulos como se describe con respecto al codificador 20 de la FIG. 2 y/o cualquier otro sistema o subsistema de codificación descrito en el presente documento. El decodificador 30 puede implementarse a través de la circuitería de procesamiento 46 para incorporar los diversos módulos como se describe con respecto al decodificador 30 de la FIG. 3 y/o cualquier otro sistema o subsistema decodificador descrito en el presente documento. La circuitería de procesamiento puede configurarse para realizar las diversas operaciones como se describe más adelante. Como se muestra en la FIG. 5, si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio de almacenamiento legible por ordenador no transitorio adecuado y puede ejecutar las instrucciones en hardware utilizando uno o más procesadores para realizar las técnicas de esta divulgación. Cualquiera de codificador de video 20 y decodificador de video 30 puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un solo dispositivo, por ejemplo, como se muestra en la FIG. 1B.

El dispositivo de origen 12 y el dispositivo de destino 14 pueden ser cualquiera de una amplia gama de dispositivos, incluyendo cualquier tipo de dispositivos de mano o fijos, por ejemplo, ordenadores portátiles, teléfonos móviles, teléfonos inteligentes, tabletas, cámaras, ordenador de sobremesa, decodificadores, televisores, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de video

(como servidores de servicios de contenidos o servidores de entrega de contenidos), dispositivos receptores de emisiones, dispositivos transmisores de emisiones, o similares, y pueden no utilizar ningún tipo de sistema operativo. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para comunicación inalámbrica. Así, el dispositivo de origen 12 y el dispositivo de destino 14 pueden ser dispositivos de comunicación inalámbrica.

En algunos casos, el sistema de codificación de video 10 ilustrado en la FIG. 1A es simplemente un ejemplo y las técnicas de la presente solicitud pueden aplicarse a configuraciones de codificación de video (por ejemplo, codificación de video o decodificación de video) que no necesariamente incluyen ninguna comunicación de datos entre los dispositivos de codificación y decodificación. En otros ejemplos, los datos se recuperan de una memoria local, se transmiten a través de una red o algo similar. Un dispositivo de codificación de video puede codificar y almacenar datos en la memoria, y/o un dispositivo de decodificación de video puede recuperar y decodificar datos de la memoria. En algunos ejemplos, la codificación y decodificación es realizada por dispositivos que no se comunican entre sí, sino que simplemente codifican datos en la memoria y/o recuperan y decodifican datos de la memoria.

Por conveniencia de la descripción, en este documento se describen realizaciones de la invención, por ejemplo, por referencia a la codificación de video de alta eficiencia (HEVC) o al software de referencia de codificación de video versátil (VVC), el estándar de codificación de video de próxima generación desarrollado por el Equipo Conjunto de colaboración en codificación de video (JCT-VC) del Grupo de expertos en codificación de video del UIT-T (VCEG) y del Grupo de expertos en imágenes en movimiento (MPEG) de ISO/IEC. Un experto en la técnica comprenderá que las realizaciones de la invención no se limitan a HEVC o VVC.

Codificador y método de codificación

La FIG. 2 muestra un diagrama de bloques esquemático de un codificador 20 de video de ejemplo que está configurado para implementar las técnicas de la presente solicitud. En el ejemplo de la FIG. 2, el codificador de video 20 comprende una entrada 201 (o interfaz de entrada 201), una unidad de cálculo residual 204, una unidad de procesamiento de transformación 206, una unidad de cuantificación 208, una unidad de cuantificación inversa 210, y una unidad de procesamiento de transformación inversa 212, una unidad de reconstrucción 214, una unidad de filtro de bucle 220, un búfer/memoria intermedia de imágenes decodificadas (DPB) 230, una unidad de selección de modo 260, una unidad de codificación de entropía 270 y una salida 272 (o interfaz de salida 272). La unidad de selección de modo 260 puede incluir una unidad de interpredicción, una unidad de intrapredicción 254 y una unidad de partición 262. La unidad de interpredicción 244 puede incluir una unidad de estimación de movimiento y una unidad de compensación de movimiento (no mostrada). Un codificador de video 20 como se muestra en la FIG. 2 también puede denominarse codificador de video híbrido o un codificador de video según un códec de video híbrido.

La unidad de cálculo residual 204, la unidad de procesamiento de transformación 206, la unidad de cuantificación 208, la unidad de selección de modo 260 puede denominarse formando una trayectoria/ruta de señal directa del codificador 20, mientras que la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 214, el búfer 216, el filtro de bucle 220, el búfer de imágenes decodificadas (DPB) 230, la unidad de interpredicción 244 y la unidad de intrapredicción 254 pueden denominarse como formando una trayectoria/ruta de señal hacia atrás del codificador de video 20, en donde la trayectoria de señal hacia atrás del codificador de video 20 corresponde a la trayectoria de señal del decodificador de video 30 en la FIG. 3). La unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 214, el filtro de bucle 220, el búfer de imágenes decodificadas (DPB) 230, la unidad de interpredicción 244 y la unidad de intrapredicción 254 también se refieren a la formación de la "Decodificador incorporado" del codificador de video 20.

Imágenes y particiones de imágenes (imágenes y bloques)

El codificador 20 puede configurarse para recibir, por ejemplo, a través de la entrada 201, una imagen 17 (o datos de imagen 17), por ejemplo, imagen de una secuencia de imágenes que forman una secuencia de video o video. La imagen o los datos de imagen recibidos también pueden ser una imagen preprocesada 19 (o datos de imagen preprocesada 19). En aras de la simplicidad, la siguiente descripción se refiere a la imagen 17. La imagen 17 también puede denominarse imagen actual o imagen a codificar (en particular en la codificación de video para distinguir la imagen actual de otras imágenes, por ejemplo, imágenes previamente codificadas y/o decodificadas de la misma secuencia de video, es decir, la secuencia de video que también comprende la imagen actual).

Una imagen (digital) es o puede considerarse como una matriz o arreglo bidimensional de muestras con valores de intensidad. La muestra en el arreglo también puede denominarse como pixel (una forma abreviada de un elemento de imagen ("picture element") o pel. El número de muestras en dirección horizontal y vertical (o eje) del arreglo o imagen define el tamaño y/o la resolución de la imagen. Para la representación del color, normalmente se emplean tres componentes de color, es decir, la imagen puede representarse o incluir tres arreglos de muestras. En el formato RGB o espacio de color, una imagen comprende un arreglo de muestras roja, verde y azul correspondiente. Sin embargo, en la codificación de video cada pixel suele representarse en un formato de luminancia y crominancia o espacio de color, por ejemplo YCbCr, que comprende un componente de luminancia indicado por Y (a veces también se utiliza L

en su lugar) y dos componentes de cromaticidad indicados por Cb y Cr. El componente de luminancia (o luma corta) Y representa el brillo o la intensidad del nivel de gris (por ejemplo, como en una imagen en escala de grises), mientras que los dos componentes de cromaticidad (o croma corta) Cb y Cr representan los componentes de información cromática o de cromaticidad. Por consiguiente, una imagen en formato YCbCr comprende un arreglo de muestras de luminancia (Y) y dos arreglos de muestras de cromaticidad (Cb y Cr). Las imágenes en formato RGB se pueden convertir o transformar en formato YCbCr y viceversa, el proceso también se conoce como transformación o conversión de color. Si una imagen es monocromática, la imagen puede comprender solo un arreglo de muestra de luminancia. En consecuencia, una imagen puede ser, por ejemplo, un arreglo de muestras de luminancia en formato monocromático o un arreglo de muestras de luminancia y dos arreglos correspondientes de muestras de cromaticidad en 4: 2: 0, 4: 2: 2 y 4: 4: 4. formato de color.

Las realizaciones del codificador de video 20 pueden comprender una unidad de partición de imágenes (no representada en la FIG. 2) configurada para particionar la imagen 17 en una pluralidad de bloques de imágenes 203 (normalmente no superpuestos). Estos bloques también pueden denominarse bloques raíz, macrobloques (H.264/AVC) o bloques de árbol de codificación (CTB) o unidades de árbol de codificación (CTU) (H.265/HEVC y VVC). La unidad de partición de imágenes puede estar configurada para utilizar el mismo tamaño de bloque para todas las imágenes de una secuencia de video y la cuadrícula correspondiente que define el tamaño de bloque, o para cambiar el tamaño de bloque entre imágenes o subconjuntos o grupos de imágenes, y particionar cada imagen en los bloques correspondientes.

En realizaciones adicionales, el codificador de video puede configurarse para recibir directamente un bloque 203 de la imagen 17, por ejemplo, uno, varios o todos los bloques que forman la imagen 17. El bloque de imagen 203 también puede denominarse bloque de imagen actual o bloque de imagen a codificar.

Al igual que la imagen 17, el bloque de imágenes 203 también es o puede considerarse como una matriz o arreglo bidimensional de muestras con valores de intensidad (valores de muestra), aunque de menor dimensión que la imagen 17. En otras palabras, el bloque 203 puede comprender, por ejemplo, un arreglo de muestra (por ejemplo, un arreglo de luma en caso de una imagen monocromática 17, o un arreglo de luma o croma en el caso de una imagen en color) o tres arreglos de muestra (por ejemplo, una luma y dos arreglos de croma en el caso de una imagen en color 17) o cualquier otro número y/o tipo de arreglos dependiendo del formato de color aplicado. El número de muestras en dirección horizontal y vertical (o eje) del bloque 203 define el tamaño del bloque 203. En consecuencia, un bloque puede, por ejemplo, un arreglo de muestras de MxN (M columnas por N filas), o un arreglo de MxN coeficientes de transformación.

Las realizaciones del codificador de video 20 como se muestra en la FIG. 2 pueden configurarse para codificar la imagen 17 bloque por bloque, por ejemplo, la codificación y predicción se realiza por bloque 203.

Las realizaciones del codificador de video 20 como se muestra en la FIG. 2 pueden configurarse adicionalmente para particionar y/o codificar la imagen usando segmentos (también denominados segmentos de video), en donde una imagen puede particionarse o codificarse usando uno o más segmentos (típicamente no superpuestos), y cada segmento puede comprender uno o más bloques (por ejemplo, CTUs).

Las realizaciones del codificador de video 20 como se muestra en la FIG. 2 pueden configurarse además para particionar y/o codificar la imagen utilizando grupos de mosaicos (también denominados grupos de mosaicos de video) y/o mosaicos (también denominados mosaicos de video), en donde una imagen puede particionarse o codificarse utilizando uno o más grupos de mosaicos (normalmente no superpuestos), y cada grupo de mosaicos puede comprender, por ejemplo, uno o más bloques (por ejemplo, CTUs) o uno o más mosaicos, en donde cada mosaico, por ejemplo, puede ser de forma rectangular y puede comprender uno o más bloques (por ejemplo, CTUs), por ejemplo, bloques completos o fraccionarios.

Cálculo residual

La unidad de cálculo residual 204 puede configurarse para calcular un bloque residual 205 (también denominado residual 205) basándose en el bloque de imagen 203 y un bloque de predicción 265 (más adelante se proporcionan más detalles sobre el bloque de predicción 265), por ejemplo, restando valores de muestra del bloque de predicción 265 de los valores de muestra del bloque de imagen 203, muestra por muestra (píxel por píxel) para obtener el bloque residual 205 en el dominio de muestra.

Transformación

La unidad de procesamiento de transformación 206 puede estar configurada para aplicar una transformada, por ejemplo una transformada coseno discreta (DCT) o una transformada seno discreta (DST), sobre los valores de muestra del bloque residual 205 para obtener los coeficientes de transformación 207 en un dominio de transformación. Los coeficientes de transformación 207 también pueden denominarse coeficientes residuales de transformación y representan el bloque residual 205 en el dominio de la transformación.

La unidad de procesamiento de transformación 206 puede estar configurada para aplicar aproximaciones enteras de DCT/DST, como las transformaciones especificadas para H.265/HEVC. En comparación con una transformada DCT ortogonal, estas aproximaciones de números enteros se escalan típicamente por un cierto factor. Para preservar la norma del bloque residual que se procesa mediante las transformaciones directa e inversa, se aplican factores de escala adicionales como parte del proceso de transformación. Los factores de escala generalmente se eligen en función de ciertas restricciones, como los factores de escala que son una potencia de dos para las operaciones de desplazamiento, la profundidad de bits de los coeficientes de transformación, el compromiso entre la precisión y los costos de implementación, etc. Los factores de escala específicos se especifican, por ejemplo, para la transformación inversa, por ejemplo mediante la unidad de procesamiento de transformación inversa 212 (y la correspondiente transformación inversa, por ejemplo, mediante la unidad de procesamiento de transformación inversa 312 en el decodificador de video 30) y los factores de escala correspondientes para la transformación directa, por ejemplo, mediante la unidad de procesamiento de transformación 206, en un codificador 20 se puede especificar en consecuencia.

Las realizaciones del codificador de video 20 (respectivamente la unidad de procesamiento de transformación 206) pueden configurarse para dar salida a parámetros de transformación, por ejemplo, un tipo de transformación o transformaciones, por ejemplo, directamente o codificado o comprimido a través de la unidad de codificación de entropía 270, de modo que, por ejemplo, el decodificador de video 30 pueda recibir y utilizar los parámetros de transformación para decodificar.

Cuantización/Cuantificación (Quantization)

La unidad de cuantificación 208 puede configurarse para cuantificar los coeficientes de transformación 207 para obtener coeficientes cuantificados 209, por ejemplo aplicando cuantificación escalar o cuantificación vectorial. Los coeficientes cuantificados 209 también pueden denominarse coeficientes de transformación cuantificados 209 o coeficientes residuales cuantificados 209.

El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes de transformación 207. Por ejemplo, un coeficiente de transformación de n bits puede redondearse a un coeficiente de transformación de m bits durante la cuantificación, cuando n es mayor que m . El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación (QP). Por ejemplo, para la cuantificación escalar, se pueden aplicar diferentes escalas para lograr una cuantificación más fina o más gruesa. Los tamaños de paso de cuantificación más pequeños corresponden a una cuantificación más fina, mientras que los tamaños de paso de cuantificación más grandes corresponden a una cuantificación más gruesa. El tamaño del paso de cuantificación aplicable puede indicarse mediante un parámetro de cuantificación (QP). El parámetro de cuantificación puede ser, por ejemplo, un índice para un conjunto predefinido de tamaños de paso de cuantificación aplicables. Por ejemplo, un parámetro de cuantificación más pequeño puede corresponder a una cuantificación más fina (un paso de cuantificación más pequeño) y un parámetro de cuantificación más grande puede corresponder a una cuantificación más gruesa (un paso de cuantificación más grande), y viceversa. La cuantificación puede incluir división por un tamaño de paso de cuantificación y una descuantificación correspondiente y/o inversa, por ejemplo, por la unidad 210 de cuantificación inversa, puede incluir la multiplicación por el tamaño del paso de cuantificación. Realizaciones de acuerdo con algunas normas, por ejemplo, HEVC, puede configurarse para usar un parámetro de cuantificación para determinar el tamaño del paso de cuantificación. Generalmente, el tamaño del paso de cuantificación puede ser calculado en base a un parámetro de cuantificación utilizando una aproximación de punto fijo de una ecuación que incluye la división. Pueden introducirse factores de escala adicionales para la cuantificación y la descuantificación a fin de restablecer la norma del bloque residual, que podría modificarse debido a la escala utilizada en la aproximación de punto fijo de la ecuación para el tamaño del paso de cuantificación y el parámetro de cuantificación. En una implementación de ejemplo, el escalado de la transformación inversa y la descuantificación podrían combinarse. Alternativamente, se pueden utilizar tablas de cuantificación personalizadas y señalarlas desde un codificador a un decodificador, por ejemplo, en un flujo de bits. La cuantificación es una operación con pérdidas, en la que la pérdida aumenta al aumentar los tamaños de los pasos de cuantificación.

Las realizaciones del codificador de video 20 (respectivamente la unidad de cuantificación 208) pueden configurarse para emitir parámetros de cuantificación (QP), por ejemplo, directamente o codificado mediante la unidad de codificación de entropía 270, de modo que, por ejemplo, el decodificador de video 30 pueda recibir y aplicar los parámetros de cuantificación para la decodificación.

Cuantificación/cuantización inversa

La unidad de cuantificación inversa 210 está configurada para aplicar la cuantificación inversa de la unidad de cuantificación 208 en los coeficientes cuantificados para obtener coeficientes descuantificados 211, por ejemplo, aplicando la inversa del esquema de cuantificación aplicado por la unidad de cuantificación 208 basándose en o usando el mismo tamaño de paso de cuantificación que la unidad de cuantificación 208. Los coeficientes descuantificados 211 también pueden denominarse coeficientes residuales descuantificados 211 y corresponden, aunque normalmente no son idénticos a los coeficientes de transformada debido a la pérdida por cuantificación, a los coeficientes de transformación 207.

Transformación inversa

5 La unidad de procesamiento de transformación inversa 212 está configurada para aplicar la transformada inversa de la transformada aplicada por la unidad de procesamiento de transformación 206, por ejemplo, una transformada coseno discreta inversa (DCT) o una transformada seno discreta inversa (DST) u otras transformadas inversas, para obtener un bloque residual reconstruido 213 (o los correspondientes coeficientes descuantificados 213) en el dominio de la muestra. El bloque residual reconstruido 213 también puede denominarse bloque de transformación 213.

10 Reconstrucción

La unidad de reconstrucción 214 (por ejemplo, el sumador o adicionador/agregador 214) está configurada para sumar el bloque de transformación 213 (es decir, el bloque residual reconstruido 213) al bloque de predicción 265 para obtener un bloque reconstruido 215 en el dominio de la muestra, por ejemplo, sumando, muestra a muestra, los valores de muestra del bloque residual reconstruido 213 y los valores de muestra del bloque de predicción 265.

Filtrado

20 La unidad de filtro de bucle 220 (o "filtro de bucle" corto 200), está configurada para filtrar el bloque 215 reconstruido para obtener un bloque filtrado 221 o, en general, para filtrar muestras reconstruidas para obtener muestras filtradas. La unidad de filtro de bucle está configurada, por ejemplo, para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del video. La unidad de filtro de bucle 220 puede comprender uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de compensación adaptable de muestra (SAO) o uno o más de otros filtros, por ejemplo, un filtro bilateral, un filtro de bucle adaptativo (ALF), un filtro de nitidez, uno de suavizado o un filtro colaborativo, o cualquier combinación de los mismos. Aunque la unidad 220 de filtro de bucle se muestra en la FIG. 2 como un filtro de bucle, en otras configuraciones, la unidad de filtro de bucle 220 puede implementarse como un filtro de bucle posterior. El bloque filtrado 221 también puede denominarse bloque reconstruido filtrado 221.

30 Las realizaciones del codificador de video 20 (respectivamente la unidad de filtro de bucle 220) pueden configurarse para emitir parámetros de filtro de bucle (tales como información de compensación adaptable de muestra), por ejemplo, directamente o codificado a través de la unidad de codificación de entropía 270, de modo que, por ejemplo, un decodificador 30 pueda recibir y aplicar los mismos parámetros de filtro de bucle o filtros de bucle respectivos para decodificar.

35 Búfer de imágenes decodificadas

El búfer/memoria intermedia de imágenes decodificadas (DPB) 230 puede ser una memoria que almacena imágenes de referencia, o en general datos de imágenes de referencia, para codificar datos de video mediante el codificador 20 de video. El DPB 230 puede ser formado por cualquiera de una variedad de dispositivos de memoria, como la memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El búfer de imágenes decodificadas (DPB) 230 puede configurarse para almacenar uno o más bloques filtrados 221. El búfer de imágenes decodificadas 230 puede estar configurado además para almacenar otros bloques previamente filtrados, por ejemplo, bloques previamente reconstruidos y filtrados 221, de la misma imagen actual o de imágenes diferentes, por ejemplo, imágenes previamente reconstruidas, y puede proporcionar imágenes completas previamente reconstruidas, es decir, decodificadas (y los correspondientes bloques de referencia y muestras) y/o una imagen actual parcialmente reconstruida (y los correspondientes bloques de referencia y muestras), por ejemplo, para la interpredicción. El búfer de imágenes decodificadas (DPB) 230 también puede configurarse para almacenar uno o más bloques 215 reconstruidos sin filtrar o, en general, muestras reconstruidas sin filtrar, por ejemplo, si el bloque 215 reconstruido no es filtrado por la unidad 220 de filtro de bucle, o cualquier otra versión procesada adicional de los bloques o muestras reconstruidos.

Selección de modo (partición y predicción)

55 La unidad 260 de selección de modo comprende la unidad 262 de partición, la unidad 244 de interpredicción y la unidad 254 de intrapredicción, y está configurada para recibir u obtener datos de imágenes originales, por ejemplo, un bloque original 203 (bloque actual 203 de la imagen actual 17), y datos de imagen reconstruida, por ejemplo, muestras o bloques reconstruidos filtrados y/o sin filtrar de la misma imagen (actual) y/o de una o una pluralidad de imágenes decodificadas previamente, por ejemplo, desde el búfer de imágenes decodificadas 230 u otras búfers (por ejemplo, búfer de línea, no mostrado). Los datos de imagen reconstruida se utilizan como datos de la imagen de referencia para la predicción, por ejemplo, interpredicción o intrapredicción, para obtener un bloque de predicción 265 o predictor 265.

65 La unidad de selección de modo 260 puede configurarse para determinar o seleccionar una partición para un modo de predicción de bloque actual (que incluye sin particiones) y un modo de predicción (por ejemplo, un modo de intra

o inter predicción) y generar un bloque 265 de predicción correspondiente, que se utiliza para la cálculo del bloque residual 205 y para la reconstrucción del bloque reconstruido 215.

Las realizaciones de la unidad de selección de modo 260 pueden configurarse para seleccionar la partición y el modo de predicción (por ejemplo, de los soportados o disponibles para la unidad de selección de modo 260), que proporcionan la mejor coincidencia o, en otras palabras, el residuo mínimo (el residuo mínimo significa mejor compresión para transmisión o almacenamiento), o una mínima sobrecarga de señalización (mínima sobrecarga de señalización significa mejor compresión para transmisión o almacenamiento), o que considera o equilibra ambos. La unidad 260 de selección de modo puede configurarse para determinar el modo de partición y predicción basándose en la optimización de la distorsión de tasa (RDO), es decir, seleccionar el modo de predicción que proporciona una distorsión de la tasa mínima. Términos como "mejor", "mínimo", "óptimo", etc. en este contexto no se refieren necesariamente a un "mejor", "mínimo", "óptimo" general, etc., sino que también pueden referirse al cumplimiento de un criterio de terminación o selección como un valor que excede o cae por debajo de un umbral u otras restricciones que conducen potencialmente a una "selección subóptima" pero que reducen la complejidad y el tiempo de procesamiento.

En otras palabras, la unidad de partición 262 puede configurarse para particionar el bloque 203 en particiones o subbloques de bloques más pequeños (que forman nuevamente bloques), por ejemplo, utilizando iterativamente la partición de árbol cuádruple (QT), la partición binaria (BT) o la partición de árbol triple (TT) o cualquier combinación de las mismas, y para realizar, por ejemplo, la predicción para cada una de las particiones o subbloques de bloque, en donde la selección de modo comprende la selección de la estructura de árbol del bloque 203 particionado y los modos de predicción se aplican a cada una de las particiones de bloque o subbloques.

A continuación, se explicará con más detalle la partición (por ejemplo, mediante la unidad de partición 260) y el procesamiento de predicción (mediante la unidad de interpredicción 244 y la unidad de intrapredicción 254) realizados por un codificador de video 20 de ejemplo.

Particionamiento

La unidad de partición 262 puede particionar (o dividir) un bloque actual 203 en particiones más pequeñas, por ejemplo, bloques más pequeños de tamaño cuadrado o rectangular. Estos bloques más pequeños (que también pueden denominarse subbloques) pueden dividirse aún más en particiones aún más pequeñas. Esto también se refiere a la partición de árboles o la partición jerárquica de árboles, en donde un bloque raíz, por ejemplo, en el nivel de árbol raíz 0 (nivel de jerarquía 0, profundidad 0), puede dividirse de forma recursiva, por ejemplo, particionado en dos o más bloques de un siguiente nivel de árbol inferior, por ejemplo, nodos en el nivel de árbol 1 (nivel de jerarquía 1, profundidad 1), en donde estos bloques pueden volver a particionarse en dos o más bloques de un nivel inferior siguiente, por ejemplo, nivel de árbol 2 (nivel de jerarquía 2, profundidad 2), etc. hasta que finalice la partición, por ejemplo, porque se cumple un criterio de terminación, por ejemplo, se alcanza una profundidad máxima de árbol o un tamaño mínimo de bloque. Los bloques que no se dividen más también se denominan bloques hoja o nodos hoja del árbol. Un árbol que utiliza la partición en dos particiones se denomina árbol binario (BT), un árbol que utiliza la partición en tres particiones se denomina árbol ternario (TT) y un árbol que utiliza la partición en cuatro particiones se denomina árbol cuádruple (QT).

Como se mencionó anteriormente, el término "bloque" como se usa en este documento puede ser una parte, en particular una parte cuadrada o rectangular, de una imagen. Con referencia, por ejemplo, para HEVC y VVC, el bloque puede ser o corresponder a una unidad de árbol de codificación (CTU), una unidad de codificación (CU), una unidad de predicción (PU) y una unidad de transformación (TU) y/o bloques correspondientes, por ejemplo, un bloque de árbol de codificación (CTB), un bloque de codificación (CB), un bloque de transformación (TB) o un bloque de predicción (PB).

Por ejemplo, una unidad de árbol de codificación (CTU) puede ser o comprender un CTB de muestras de luma, dos CTBs correspondientes de muestras de croma de una imagen que tiene tres arreglos de muestras, o un CTB de muestras de una imagen monocromática o una imagen que está codificada utilizando tres planos de color separados y estructuras de sintaxis utilizadas para codificar las muestras. En consecuencia, un bloque de árbol de codificación (CTB) puede ser un bloque de NxN muestras para algún valor de N, de modo que la división de un componente en CTB es una partición. Una unidad de codificación (CU) puede ser o comprender un bloque de codificación de muestras de luma, dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene tres arreglos de muestras, o un bloque de codificación de muestras de una imagen monocromática o una imagen que se codifica usando tres planos de color separados y estructuras de sintaxis que se utilizan para codificar las muestras. De manera correspondiente, un bloque de codificación (CB) puede ser un bloque de MxN muestras para algunos valores de M y N, de manera que la división de un CTB en bloques de codificación es una partición.

En realizaciones, por ejemplo, de acuerdo con HEVC, una unidad de árbol de codificación (CTU) puede dividirse en CUs utilizando una estructura de árbol cuádruple indicada como árbol de codificación. La decisión de codificar un área de imagen usando predicción inter-imagen (temporal) o intra-imagen (espacial) se toma a nivel de CU. Cada CU se puede dividir más en una, dos o cuatro PUs según un tipo de división de PU. Dentro de una PU, se aplica el mismo

proceso de predicción y la información relevante se transmite al decodificador por PU. Después de obtener el bloque residual aplicando el proceso de predicción basado en el tipo de división de la PU, una CU puede dividirse en unidades de transformación (TU) según otra estructura de árbol cuádruple similar al árbol de codificación de la CU.

En realizaciones, por ejemplo, de acuerdo con el último estándar de codificación de video actualmente en desarrollo, que se conoce como codificación de video versátil (VVC), una partición combinada de árbol cuádruple y árbol binario (QTBT) es, por ejemplo, usada para particionar un bloque de codificación. En la estructura de bloques de QTBT, una CU puede tener forma cuadrada o rectangular. Por ejemplo, una unidad de árbol de codificación (CTU) se divide primero en una estructura de árbol cuádruple. Los nodos hoja de árbol cuádruple se dividen aún más mediante un árbol binario o una estructura de árbol ternario (o triple). Los nodos hoja del árbol de partición se denominan unidades de codificación (CUs), y esa segmentación se utiliza para la predicción y el procesamiento de transformación sin más particiones. Esto significa que CU, PU y TU tienen el mismo tamaño de bloque en la estructura del bloque de codificación de QTBT. En paralelo, la partición múltiple, por ejemplo, la partición de árbol triple puede ser utilizada junto con la estructura de bloques de QTBT.

En un ejemplo, la unidad de selección de modo 260 del codificador de video 20 puede estar configurada para realizar cualquier combinación de las técnicas de partición descritas en el presente documento.

Como se describió anteriormente, el codificador de video 20 está configurado para determinar o seleccionar un modo de predicción mejor u óptimo para un conjunto de modos de predicción (por ejemplo, pre-determinados). El conjunto de modos de predicción puede comprender, por ejemplo, modos de intrapredicción y/o modos de interpredicción.

Intrapredicción

El conjunto de modos de intrapredicción puede comprender 35 modos de intrapredicción diferentes, por ejemplo, modos no direccionales como el modo DC (o medio) y el modo planar, o modos direccionales, por ejemplo, como se define en HEVC, o puede comprender 67 modos de intrapredicción diferentes, por ejemplo, modos no direccionales como el modo DC (o medio) y el modo planar, o modos direccionales, por ejemplo, como se define en VVC.

La unidad de intrapredicción 254 está configurada para usar muestras reconstruidas de bloques vecinos de la misma imagen actual para generar un bloque de intrapredicción 265 de acuerdo con un modo de intrapredicción del conjunto de modos de intrapredicción.

La unidad de intrapredicción 254 (o en general la unidad de selección de modo 260) está configurada además para emitir parámetros de intrapredicción (o en general información indicativa del modo de intrapredicción seleccionado para el bloque) a la unidad de codificación de entropía 270 en forma de elementos de sintaxis 266 para su inclusión en los datos de imagen codificada 21, de modo que, por ejemplo, el decodificador de video 30 puede recibir y utilizar los parámetros de predicción para la descodificación.

Inter-predicción

El conjunto de (o posibles) modos de interpredicción depende de las imágenes de referencia disponibles (es decir, imágenes previas al menos parcialmente decodificadas, por ejemplo, almacenadas en DBP 230) y otros parámetros de interpredicción, por ejemplo, si la imagen de referencia completa o solo una parte, por ejemplo, un área de ventana de búsqueda alrededor del área del bloque actual, de la imagen de referencia se usa para buscar un bloque de referencia que coincida mejor, y/o, por ejemplo, si se aplica la interpolación de píxeles, por ejemplo, interpolación de medio/semi-pel y/o cuarto de pel, o no.

Además de los modos de predicción anteriores, se puede aplicar el modo de salto y/o el modo directo.

La unidad de interpredicción 244 puede incluir una unidad de estimación de movimiento (ME) y una unidad de compensación de movimiento (MC) (ambas no mostradas en la FIG. 2). La unidad de estimación de movimiento puede estar configurada para recibir u obtener el bloque de imagen 203 (bloque de imagen actual 203 de la imagen actual 17) y una imagen decodificada 231, o al menos uno o una pluralidad de bloques previamente reconstruidos, por ejemplo, bloques reconstruidos de una o una pluralidad de otras/diferentes imágenes previamente decodificadas 231, para la estimación de movimiento. Por ejemplo, una secuencia de video puede comprender la imagen actual y las imágenes previamente decodificadas 231, o en otras palabras, la imagen actual y las imágenes previamente decodificadas 231 pueden ser parte o formar una secuencia de imágenes que forman una secuencia de video.

El codificador 20 puede, por ejemplo, estar configurado para seleccionar un bloque de referencia de entre una pluralidad de bloques de referencia de la misma o diferentes imágenes de la pluralidad de otras imágenes y proporcionar una imagen de referencia (o índice de imagen de referencia) y/o un desplazamiento (desplazamiento espacial) entre la posición (coordenadas x, y) del bloque de referencia y la posición del bloque actual como parámetros de interpredicción a la unidad de estimación de movimiento. Este desplazamiento también se denomina vector de movimiento (MV).

La unidad de compensación de movimiento está configurada para obtener, por ejemplo, recibir, un parámetro de interpredicción y realizar una interpredicción basada en o utilizando el parámetro de interpredicción para obtener un bloque de interpredicción 265. La compensación de movimiento, realizada por la unidad de compensación de movimiento, puede implicar la obtención o generación del bloque de predicción basado en el vector de movimiento/bloque determinado por la estimación de movimiento, posiblemente realizando interpolaciones con precisión sub-píxel. El filtrado de interpolación puede generar muestras de píxeles adicionales a partir de muestras de píxeles conocidas, aumentando así potencialmente el número de bloques de predicción candidatos que pueden utilizarse para codificar un bloque de imágenes. Al recibir el vector de movimiento para el PU del bloque de imágenes actual, la unidad de compensación de movimiento puede localizar el bloque de predicción al que apunta el vector de movimiento en una de las listas de imágenes de referencia.

La unidad de compensación de movimiento también puede generar elementos de sintaxis asociados a los bloques y a los segmentos para su uso por el decodificador de video 30 en la decodificación de los bloques de imagen del segmento de video. Además o como una alternativa a los segmentos y los elementos de sintaxis respectivos, se pueden generar o utilizar grupos de mosaicos y/o mosaicos y los elementos de sintaxis respectivos.

Codificación de entropía

La unidad de codificación de entropía 270 está configurada para aplicar, por ejemplo, un algoritmo o esquema de codificación de entropía (por ejemplo, un esquema de codificación de longitud variable (VLC), un esquema de VLC adaptable al contexto (CAVLC), un esquema de codificación aritmética, una binarización, un esquema de codificación de longitud variable (VLC) codificación aritmética binaria (CABAC), codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), codificación de entropía de partición de intervalo de probabilidad (PIPE) u otra metodología o técnica de codificación de entropía) o derivar (sin compresión) en los coeficientes cuantificados 209, parámetros de interpredicción, parámetros de intrapredicción, parámetros de filtro de bucle y/u otros elementos de sintaxis para obtener datos de imagen codificados 21 que se pueden emitir a través de la salida 272, por ejemplo, en forma de un flujo de bits codificado 21, de modo que, por ejemplo, el decodificador de video 30 pueda recibir y utilizar los parámetros para decodificar. El flujo de bits codificado 21 puede ser transmitido al decodificador de video 30, o almacenado en una memoria para su posterior transmisión o recuperación por el decodificador de video 30.

Se pueden utilizar otras variaciones estructurales del codificador de video 20 para codificar el flujo de video. Por ejemplo, un codificador no basado en la transformación 20 puede cuantificar la señal residual directamente sin la unidad de procesamiento de la transformación 206 para ciertos bloques o fotogramas. En otra implementación, un codificador 20 puede tener la unidad de cuantificación 208 y la unidad de cuantificación inversa 210 combinadas en una sola unidad.

Método de decodificación y decodificación

La FIG. 3 muestra un ejemplo de un decodificador de video 30 que está configurado para implementar las técnicas de la presente solicitud. El decodificador de video 30 está configurado para recibir los datos de imagen codificados 21 (por ejemplo, el flujo de bits codificado 21), por ejemplo, codificados por el codificador 20, para obtener una imagen decodificada 331. Los datos de imagen codificados o el flujo de bits comprenden información para decodificar los datos de imagen codificados, por ejemplo, datos que representan bloques de imágenes de un segmento de video decodificado (y/o grupos de mosaicos o mosaicos) y elementos de sintaxis asociados.

En el ejemplo de la FIG. 3, el decodificador 30 comprende una unidad de decodificación de entropía 304, una unidad de cuantificación inversa 310, una unidad de procesamiento de transformación inversa 312, una unidad de reconstrucción 314 (por ejemplo, un sumador 314), un filtro de bucle 320, un búfer de imágenes decodificadas (DBP) 330, una unidad de aplicación de modo 360, una unidad de interpredicción 344 y una unidad de intrapredicción 354. La unidad de interpredicción 344 puede ser o incluir una unidad de compensación de movimiento. El decodificador de video 30 puede, en algunos ejemplos, realizar una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de video 100 de la FIG. 2.

Como se explica con respecto al codificador 20, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 214, el filtro de bucle 220, el búfer de imágenes decodificadas (DPB) 230, la unidad de interpredicción 344 y la unidad de intrapredicción 354 también se denominan como formando el "decodificador incorporado" del codificador de video 20. En consecuencia, la unidad de cuantificación inversa 310 puede ser idéntica en función a la unidad de cuantificación inversa 110, la unidad de procesamiento de transformación inversa 312 puede ser idéntica en función a la unidad de procesamiento de transformación inversa 212, la unidad de reconstrucción 314 puede ser idéntica en función a la unidad de reconstrucción 214, el filtro de bucle 320 puede ser idéntico en función al filtro de bucle 220, y el búfer de imágenes decodificadas 330 puede ser idéntica en función al búfer de imágenes decodificadas 230. Por lo tanto, las explicaciones proporcionadas para las respectivas unidades y funciones del codificador de video 20 se aplican correspondientemente a las respectivas unidades y funciones del decodificador de video 30.

Decodificación de entropía

La unidad de decodificación de entropía 304 está configurada para analizar el flujo de bits 21 (o en general datos de imagen codificados 21) y realizar, por ejemplo, decodificación de entropía a los datos de imagen codificados 21 para obtener, por ejemplo, coeficientes cuantificados 309 y/o parámetros de codificación decodificados (no se muestra en la FIG.3), por ejemplo, cualquiera o todos los parámetros de interpretación (por ejemplo, índice de imagen de referencia y vector de movimiento), parámetro de intrapredicción (por ejemplo, modo o índice de intrapredicción), parámetros de transformación, parámetros de cuantificación, parámetros de filtro de bucle y/u otros elementos de sintaxis. La unidad 304 de decodificación de entropía puede configurarse para aplicar los algoritmos o esquemas de decodificación correspondientes a los esquemas de codificación descritos con respecto a la unidad 270 de codificación de entropía del codificador 20. La unidad de decodificación de entropía 304 puede configurarse además para proporcionar parámetros de interpretación, parámetro de intrapredicción y/u otros elementos de sintaxis a la unidad de aplicación de modo 360 y otros parámetros a otras unidades del decodificador 30. El decodificador de video 30 puede recibir los elementos de sintaxis en el nivel de segmento de video y/o en el nivel de bloque de video. Además o como alternativa a los segmentos y los elementos de sintaxis respectivos, se pueden recibir y/o utilizar grupos de mosaicos y/o mosaicos y los elementos de sintaxis respectivos.

Cuantificación/cuantización inversa

La unidad de cuantificación inversa 310 puede configurarse para recibir parámetros de cuantificación (QP) (o en general información relacionada con la cuantificación inversa) y coeficientes cuantificados de los datos de imagen codificados 21 (por ejemplo, mediante análisis sintáctico ("parsing") y/o decodificación, por ejemplo, mediante la unidad de decodificación de entropía 304) y aplicar en base a los parámetros de cuantificación una cuantificación inversa sobre los coeficientes cuantificados decodificados 309 para obtener coeficientes descuantificados 311, que también pueden denominarse coeficientes de transformación 311. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación determinado por el codificador de video 20 para cada bloque de video en el segmento de video (o mosaico o grupo de mosaicos) para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debe aplicarse.

Transformación inversa

La unidad de procesamiento de transformación inversa 312 puede configurarse para recibir coeficientes descuantificados 311, también denominados coeficientes de transformación 311, y para aplicar una transformación a los coeficientes descuantificados 311 para obtener bloques residuales reconstruidos 213 en el dominio de muestra. Los bloques residuales reconstruidos 213 también pueden denominarse bloques de transformación 313. La transformación puede ser una transformación inversa, por ejemplo, una DCT inversa, una DST inversa, una transformada de entero inverso o un proceso de transformación inversa conceptualmente similar. La unidad de procesamiento de transformación inversa 312 puede configurarse además para recibir parámetros de transformación o información correspondiente de los datos de imagen codificados 21 (por ejemplo, analizando sintácticamente y/o decodificando, por ejemplo, mediante la unidad de decodificación de entropía 304) para determinar la transformación que se aplicará a los coeficientes descuantificados 311.

Reconstrucción

La unidad de reconstrucción 314 (por ejemplo, sumador o adicionador/agregador 314) puede estar configurada para añadir el bloque residual reconstruido 313, al bloque de predicción 365 para obtener un bloque reconstruido 315 en el dominio de la muestra, por ejemplo, sumando los valores de muestra del bloque residual reconstruido 313 y los valores de muestra del bloque de predicción 365.

Filtrado

La unidad de filtro de bucle 320 (ya sea en el bucle de codificación o después del bucle de codificación) está configurada para filtrar el bloque reconstruido 315 para obtener un bloque filtrado 321, por ejemplo, para suavizar las transiciones de píxeles, o para mejorar de otro modo la calidad del video. La unidad de filtro de bucle 320 puede comprender uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de compensación adaptable de muestra (SAO) o uno o más de otros filtros, por ejemplo, un filtro bilateral, un filtro de bucle adaptativo (ALF), un filtro de nitidez, uno de suavizado o un filtro colaborativo, o cualquier combinación de los mismos. Aunque la unidad 320 de filtro de bucle se muestra en la FIG. 3 como un filtro de bucle, en otras configuraciones, la unidad de filtro de bucle 320 puede implementarse como un filtro de bucle posterior.

Búfer de imágenes decodificadas

Los bloques de video decodificados 321 de una imagen se almacenan luego en el búfer de imágenes decodificadas 330, que almacena las imágenes decodificadas 331 como imágenes de referencia para la compensación de movimiento subsiguiente para otras imágenes y/o para la visualización de salida respectivamente.

El decodificador 30 está configurado para emitir la imagen decodificada 311, por ejemplo, a través de la salida 312, para presentación o visualización a un usuario.

Predicción

La unidad de interpredicción 344 puede ser idéntica a la unidad de interpredicción 244 (en particular a la unidad de compensación de movimiento) y la unidad de intrapredicción 354 puede ser idéntica a la unidad de interpredicción 254 en función, y realiza decisiones de división o partición y predicciones basadas en los parámetros de partición y/o predicción o la información respectiva recibida de los datos de imagen codificada 21 (por ejemplo, mediante análisis sintáctico y/o decodificación, por ejemplo, mediante la unidad de decodificación de entropía 304). La unidad de aplicación de modo 360 puede configurarse para realizar la predicción (intra o inter predicción) por bloque basándose en imágenes reconstruidas, bloques o muestras respectivas (filtradas o sin filtrar) para obtener el bloque de predicción 365.

Cuando el segmento de video se codifica como un segmento intracodificada (I), la unidad de intrapredicción 354 de la unidad de aplicación de modo 360 está configurada para generar el bloque de predicción 365 para un bloque de imagen del segmento de video actual basándose en un modo de intrapredicción señalado y en los datos de los bloques previamente decodificados de la imagen actual. Cuando la imagen de video se codifica como un segmento intercodificado (es decir, B, o P), la unidad de interpredicción 344 (por ejemplo, la unidad de compensación de movimiento) de la unidad de aplicación de modo 360 está configurada para producir bloques de predicción 365 para un bloque de video del segmento de video actual basándose en los vectores de movimiento y otros elementos de sintaxis recibidos de la unidad de decodificación de entropía 304. Para la interpredicción, los bloques de predicción pueden producirse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de video 30 puede construir las listas de fotogramas de referencia, la Lista 0 y la Lista 1, utilizando técnicas de construcción predeterminadas basadas en las imágenes de referencia almacenadas en el DPB 330. Se puede aplicar lo mismo o similar para o mediante realizaciones que usan grupos de mosaicos (por ejemplo, grupos de mosaicos de video) y/o mosaicos (por ejemplo, mosaicos de video) además o alternativamente a los segmentos (por ejemplo, segmentos de video), por ejemplo, un video se puede codificar usando grupos de mosaicos y/o mosaicos I, P o B.

La unidad de aplicación de modo 360 está configurada para determinar la información de predicción para un bloque de video de la porción de video actual analizando los vectores de movimiento o información relacionado y otros elementos de sintaxis, y utiliza la información de predicción para producir los bloques de predicción para el bloque de video actual que se está decodificando. Por ejemplo, la unidad de aplicación de modo 360 utiliza algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra o inter predicción) utilizado para codificar los bloques de video de la porción de video, un tipo de porción de interpredicción (por ejemplo, segmento B, segmento P, o segmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de video intercodificado del segmento, estado de interpredicción para cada bloque de video intercodificado del segmento, y otra información para decodificar los bloques de video en el segmento de video actual. Se puede aplicar lo mismo o similar para o mediante realizaciones que usan grupos de mosaicos (por ejemplo, grupos de mosaicos de video) y/o mosaicos (por ejemplo, mosaicos de video) además o alternativamente a los segmentos (por ejemplo, segmentos de video), por ejemplo, un video se puede codificar usando grupos de mosaicos y/o mosaicos I, P o B.

Las realizaciones del decodificador de video 30 como se muestra en la FIG. 3 pueden configurarse adicionalmente para particionar y/o decodificar la imagen usando segmentos (también denominados segmentos de video), en donde una imagen puede particionarse o decodificarse usando uno o más segmentos (típicamente no superpuestos), y cada segmento puede comprender uno o más bloques (por ejemplo, CTUs).

Las realizaciones del decodificador de video 30 como se muestra en la FIG. 3 pueden configurarse para particionar y/o decodificar la imagen utilizando grupos de mosaicos (también denominados grupos de mosaicos de video) y/o mosaicos (también denominados mosaicos de video), en donde una imagen puede particionarse o decodificarse utilizando uno o más grupos de mosaicos (normalmente no superpuestos), y cada grupo de mosaicos puede comprender, por ejemplo, uno o más bloques (por ejemplo, CTUs) o uno o más mosaicos, en donde cada mosaico, por ejemplo, puede ser de forma rectangular y puede comprender uno o más bloques (por ejemplo, CTUs), por ejemplo, bloques completos o fraccionarios.

Se pueden usar otras variaciones del decodificador de video 30 para decodificar los datos de imagen codificados 21. Por ejemplo, el decodificador 30 puede producir el flujo de video de salida sin la unidad de filtrado de bucle 320. Por ejemplo, un decodificador 30 basado en no-transformación puede cuantificar inversamente la señal residual directamente sin la unidad de procesamiento de transformación inversa 312 para ciertos bloques o fotogramas. En otra implementación, el decodificador de video 30 puede tener la unidad de cuantificación inversa 310 y la unidad de procesamiento de transformación inversa 312 combinadas en una sola unidad.

Debe entenderse que, en el codificador 20 y el decodificador 30, un resultado de procesamiento de un paso actual puede procesarse adicionalmente y luego enviarse al siguiente paso. Por ejemplo, después del filtrado de

interpolación, la derivación del vector de movimiento o el filtrado de bucle, se puede realizar una operación adicional, como recortar o desplazar, sobre el resultado del procesamiento del filtrado de interpolación, la derivación del vector de movimiento o el filtrado de bucle.

Cabe señalar que se pueden aplicar operaciones adicionales a los vectores de movimiento derivados del bloque actual (incluidos, entre otros, los vectores de movimiento puntuales del modo afín, los vectores de movimiento del subbloque en los modos afín, plano/planar, ATMVP, vectores de movimiento temporal, etc.). Por ejemplo, el valor del vector de movimiento está restringido a un rango predefinido de acuerdo con su bit de representación. Si el bit que representa el vector de movimiento es bitDepth, entonces el rango es $-2^{(\text{bitDepth}-1)} \sim 2^{(\text{bitDepth}-1)}-1$, donde “^” significa exponenciación. Por ejemplo, si bitDepth se establece en 16, el rango es -32768 ~ 32767; si bitDepth se establece en 18, el rango es -131072 ~ 131071. Por ejemplo, el valor del vector de movimiento derivado (por ejemplo, los MVs de cuatro subbloques de 4x4 dentro de un bloque de 8x8) está restringido de modo que la diferencia máxima entre las partes enteras de los cuatro MVs del subbloque de 4x4 no supere los N píxeles, como no más de 1 píxel. Aquí se proporcionan dos métodos para restringir el vector de movimiento según el bitDepth.

Método 1: eliminar el MSB de desbordamiento (bit más significativo) mediante operaciones de flujo

$$ux = (mvx + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (1)$$

$$mvx = (ux \geq 2^{\text{bitDepth}-1}) ? (ux - 2^{\text{bitDepth}}) : ux \quad (2)$$

$$uy = (mvy + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (3)$$

$$mvy = (uy \geq 2^{\text{bitDepth}-1}) ? (uy - 2^{\text{bitDepth}}) : uy \quad (4)$$

donde mvx es un componente horizontal de un vector de movimiento de un bloque de imagen o un subbloque, mvy es un componente vertical de un vector de movimiento de un bloque de imagen o un subbloque, y ux y uy indican un valor intermedio;

Por ejemplo, si el valor de mvx es -32769, después de aplicar la fórmula (1) y (2), el valor resultante es 32767. En el sistema informático, los números decimales se almacenan como complemento a dos. El complemento a dos de -32769 es 1,0111,1111,1111,1111 (17 bits), luego se descarta el MSB, por lo que el complemento a dos resultante es 0111,1111,1111,1111 (el número decimal es 32767), que es el mismo como la salida aplicando la fórmula (1) y (2).

$$ux = (mvp_x + mvd_x + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (5)$$

$$mvx = (ux \geq 2^{\text{bitDepth}-1}) ? (ux - 2^{\text{bitDepth}}) : ux \quad (6)$$

$$uy = (mvp_y + mvd_y + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (7)$$

$$mvy = (uy \geq 2^{\text{bitDepth}-1}) ? (uy - 2^{\text{bitDepth}}) : uy \quad (8)$$

Las operaciones se pueden aplicar durante la suma de mvp y mvd, como se muestra en la fórmula (5) a (8).

Método 2: eliminar el MSB de desbordamiento recortando el valor

$$vx = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1}-1, vx)$$

$$vy = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1}-1, vy)$$

donde vx es un componente horizontal de un vector de movimiento de un bloque o subbloque de imagen, vy es un componente vertical de un vector de movimiento de un bloque o subbloque de imagen; x, y y z corresponden respectivamente a tres valores de entrada del proceso de recorte de MV, y la definición de la función Clip3 es la siguiente:

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{de otra forma} \end{cases}$$

La FIG. 4 es un diagrama esquemático de un dispositivo de codificación de video 400 según una realización de la divulgación. El dispositivo de codificación de video 400 es adecuado para implementar las realizaciones divulgadas tal y como se describen en el presente documento. En una realización, el dispositivo de codificación de video 400 puede ser un decodificador como el decodificador de video 30 de la FIG. 1A o un codificador como el codificador de video 20 de la FIG. 1A.

El dispositivo de codificación de video 400 comprende puertos de entrada 410 (o puertos de entrada 410) y unidades receptoras (Rx) 420 para recibir datos; un procesador, unidad lógica o unidad central de procesamiento (CPU) 430 para procesar los datos; unidades transmisoras (Tx) 440 y puertos de egreso 450 (o puertos de salida 450) para transmitir los datos; y una memoria 460 para almacenar los datos. El dispositivo de codificación de video 400 también puede comprender componentes ópticos-eléctricos (OE) y componentes eléctricos-ópticos (EO) acoplados a los puertos de entrada 410, las unidades receptoras 420, las unidades transmisoras 440 y los puertos de salida 450 para la salida o entrada de señales ópticas o eléctricas.

El procesador 430 se implementa mediante hardware y software. El procesador 430 puede implementarse como uno o más chips de CPU, núcleos (por ejemplo, como un procesador multinúcleo), FPGAs, ASICs y DSPs. El procesador 430 está en comunicación con los puertos de entrada 410, las unidades receptoras 420, las unidades transmisoras 440, los puertos de salida 450 y la memoria 460. El procesador 430 comprende un módulo de codificación 470. El módulo de codificación 470 implementa las realizaciones divulgadas anteriormente descritas. Por ejemplo, el módulo de codificación 470 implementa, procesa, prepara o proporciona las diversas operaciones de codificación. La inclusión del módulo de codificación 470 proporciona por lo tanto una mejora sustancial a la funcionalidad del dispositivo de codificación de video 400 y efectúa una transformación del dispositivo de codificación de video 400 a un estado diferente. Alternativamente, el módulo de codificación 470 se implementa como instrucciones almacenadas en la memoria 460 y ejecutadas por el procesador 430.

La memoria 460 puede comprender uno o más discos, unidades de cinta y unidades de estado sólido, y puede utilizarse como dispositivo de almacenamiento de datos de sobrecarga, para almacenar programas cuando dichos programas se seleccionan para su ejecución, y para almacenar instrucciones y datos que se leen durante la ejecución del programa. La memoria 460 puede ser, por ejemplo, volátil y/o no volátil y puede ser una memoria de sólo lectura (ROM), una memoria de acceso aleatorio (RAM), una memoria ternaria de contenido direccionable (TCAM), y/o una memoria estática de acceso aleatorio (SRAM).

La FIG. 5 es un diagrama de bloques simplificado de un aparato 500 que puede utilizarse como uno o ambos dispositivos de origen 12 y de destino 14 de la FIG. 1 según una realización ejemplar.

Un procesador 502 en el aparato 500 puede ser una unidad central de procesamiento. Alternativamente, el procesador 502 puede ser cualquier otro tipo de dispositivo, o múltiples dispositivos, capaces de manipular o procesar información existente ahora o desarrollada en el futuro. Aunque las implementaciones divulgadas pueden practicarse con un solo procesador, como se muestra, por ejemplo, el procesador 502, pueden lograrse ventajas de velocidad y eficiencia utilizando más de un procesador.

Una memoria 504 en el aparato 500 puede ser un dispositivo de memoria de sólo lectura (ROM) o un dispositivo de memoria de acceso aleatorio (RAM) en una implementación. Como memoria 504 puede utilizarse cualquier otro tipo de dispositivo de almacenamiento adecuado. La memoria 504 puede incluir código y datos 506 a los que accede el procesador 502 mediante un bus 512. La memoria 504 puede incluir además un sistema operativo 508 y programas de aplicación 510, los programas de aplicación 510 incluyendo al menos un programa que permite al procesador 502 realizar los métodos descritos aquí. Por ejemplo, los programas de aplicación 510 pueden incluir las aplicaciones 1 a N, que además incluyen una aplicación de codificación de video que realiza los métodos aquí descritos.

El aparato 500 también puede incluir uno o más dispositivos de salida, como una pantalla 518. La pantalla 518 puede ser, en un ejemplo, una pantalla sensible al tacto que combina una pantalla con un elemento sensible al tacto que es operable para detectar entradas táctiles. La pantalla 518 puede estar acoplada al procesador 502 a través del bus 512.

Aunque se muestra aquí como un solo bus, el bus 512 del aparato 500 puede estar compuesto por múltiples buses. Además, el almacenamiento secundario 514 puede estar directamente acoplado a los otros componentes del aparato 500 o se puede acceder a él a través de una red y puede comprender una única unidad integrada, como una tarjeta de memoria, o varias unidades, como múltiples tarjetas de memoria. Por tanto, el aparato 500 puede implementarse en una amplia variedad de configuraciones.

Antecedente de Codificación de modo de intrapredicción

En el VTM 4.0 actual, se construye una lista de 6-MPM (modos más probables) para la codificación del modo de intrapredicción. La lista MPM se introduce para reducir los bits necesarios para codificar el modo de intrapredicción del bloque actual. La lista de 6-MPM se construye basándose en modos de intrapredicción de bloques vecinos del bloque actual. Cuando el modo de intrapredicción del bloque actual cae en la lista MPM, se codifica un índice en lugar del modo real, que puede usar menos bits. Cuando el modo de intrapredicción del bloque actual no cae en la lista de MPM, la binarización truncada se usa para codificar el modo de intrapredicción del bloque actual. La lista MPM proporciona una buena predicción para el modo de intrapredicción del bloque actual.

Lista MPM para línea de referencia múltiple (MRL).

En VTM 4.0, la herramienta de codificación de línea de referencia múltiple (MRL) puede usar una de las múltiples líneas de referencia vecinas para predecir las muestras del bloque actual. Cuando un valor de índice de línea MRL igual a 0 (es decir, se usa la línea de referencia vecina más cercana, es decir, la línea de referencia 0 como se muestra en la FIG. 6), se usa una lista normal de 6 MPM que contiene planar y DC. Cuando un valor del índice de línea MRL no es igual a 0, se utiliza una lista de 6-MPM excluyendo planar (es decir, un valor de 0) y DC (es decir, un valor de 1).

Lista de MPM para el modo de codificación de intra subpartición (ISP)

El modo de codificación de intra subpartición (ISP) es una herramienta recientemente adoptada en VTM4.0 (JVET-M0102). Los bloques de luma intra-predichos se dividen vertical u horizontalmente en 2 o 4 subparticiones, según las dimensiones del tamaño del bloque, como se muestra en la Tabla 1. La FIG. 7 y la FIG. 8 muestran ejemplos de las dos posibilidades. Todas las subparticiones cumplen la condición de tener al menos 16 muestras.

Tabla1: Número de subparticiones según el tamaño del bloque

Tamaño de bloque	Número de subparticiones
4 × 4	No dividido
4 × 8 y 8 × 4	2
Todos los otros casos	4

Cuando se habilita el modo de codificación de intra subpartición, en VTM 4.0 se usa otra lista MPM que excluye el modo DC. El modo de intra subpartición se puede aplicar cuando un valor de índice de línea de referencia múltiple es igual a 0 (es decir, no se aplica MRL al bloque de intrapredicción actual). Todas las subparticiones comparten un único modo de intrapredicción, por lo tanto, la lista MPM se construye una vez para el intra bloque y es compartida por todas las subparticiones.

La construcción de MPM podría depender del modo de división de ISP. Se determinan dos modos de división: horizontal o vertical. Se muestran como en la FIG. 7 y la FIG. 8, donde la división/partición horizontal se muestra en la parte superior y la división/partición vertical se muestra en la parte inferior.

La siguiente Tabla 2 resume las características de la lista de 3 MPM utilizada en VTM 4.0.

Tabla 2: diferentes 6-MPM utilizados en VTM 4.0 en diferentes circunstancias

Configuración de MRL	Configuración de ISP	Característica de lista MPM
$MRL \neq 0$	-	6 MPM pero excluyendo el modo planar y el modo DC
$MRL == 0$	1	6 MPM pero excluyendo el modo DC
$MRL == 0$	0	6 MPM que incluye modo planar y modo DC

Antecedentes sobre el modelado/codificación de contexto CABAC.

Como se muestra en la Figura 9, la codificación CABAC se compone de binarización, modelado de contexto y codificación aritmética binaria. La binarización mapea/asigna los elementos de sintaxis a los símbolos binarios ("bins"). El modelado de contexto estima la probabilidad de cada bin no omitido (es decir, codificado regularmente) en base a algún contexto específico. Finalmente, la codificación aritmética binaria comprime los bins en bits de acuerdo con la probabilidad estimada.

Al descomponer cada valor de elemento de sintaxis no binario en una secuencia de bins, el procesamiento posterior de cada valor de bin en CABAC depende de la decisión del modo de codificación asociado, que puede elegirse como modo regular o de derivación ("bypass"). Este último se elige para bins, que se supone que están distribuidos uniformemente y para los cuales, en consecuencia, todo el proceso de codificación (y decodificación) aritmética binaria regular simplemente se deriva/se pasa por alto. En el modo de codificación regular, cada valor de bin se codifica utilizando el motor de codificación aritmética binaria regular, donde el modelo de probabilidad asociado se determina mediante una elección fija, según el tipo de elemento de sintaxis y la posición del bin o el índice de bin (binIdx) en la representación binarizada del elemento de sintaxis, o elegido de forma adaptable a partir de dos o más modelos de probabilidad dependiendo de la información complementaria relacionada.

La selección del modelo de probabilidad se conoce como modelado de contexto. Cada modelo de probabilidad en CABAC se aborda utilizando un índice de contexto único (ctxIdx), ya sea determinado por una asignación fija o calculado por la lógica de derivación de contexto mediante la cual, a su vez, se especifica el modelo de contexto dado.

Antecedentes sobre cómo determinar los vecinos izquierdos y superiores.

La codificación de modo de intrapredicción de VVC depende de los modos de intrapredicción de sus bloques vecinos. Es decir, los bloques izquierdo y superior del bloque actual. Su posición se muestra en la FIG. 10.

Señalización del índice MPM

Se utiliza una bandera (es decir, bandera MPM) para indicar si el modo de intrapredicción del bloque actual está en la lista MPM o no. Cuando la bandera MPM es verdadera (es decir, un valor de 1), entonces el modo de intrapredicción puede determinarse usando un índice MPM. El índice MPM se codifica utilizando un código unario truncado, como se muestra en la siguiente Tabla 3 cuando la longitud de la lista MPM es 6. Cuando la bandera MPM no es verdadero, el modo de intrapredicción del bloque actual no está en la lista MPM y el modo se codifica utilizando código binario truncado. Otro ejemplo de índice MPM cuando la lista MPM se compone de 5 modos se muestra en la Tabla 4. La cadena de bin para codificar el índice se denomina código unario truncado (no código binario truncado para codificar modos no MPM). Tenga en cuenta que para el código unario truncado, el bin 0 y 1 son intercambiables sin pérdida de generalidad. En un ejemplo, el índice MPM cuando la lista MPM se compone de 5 modos también se puede codificar como se muestra en la Tabla 5.

Tabla 3 Bandera MPM y codificación de índice MPM cuando hay 6 entradas en una lista MPM

Modos de intrapredicción	Bandera MPM	Índice MPM	Cadena de bin de índice MPM
Modos de MPM (6)	1	0	0
		1	10
		2	110
		3	1110
		4	11110
		5	11111
	0		Código binario truncado (TB)

Tabla 4 Bandera MPM y codificación de índice MPM, cuando hay 5 entradas en una lista MPM

Modos de intrapredicción	Bandera MPM	Índice MPM	Cadena de bin de índice MPM
Modos de MPM (5)	1	0	0
		1	10
		2	110
		3	1110
		4	1111
	0		Código binario truncado (TB)

Tabla 5 Bandera MPM y codificación de índice MPM, cuando hay 5 entradas en una lista de MPM y palabra de código alternativa para el índice MPM

Modos de intrapredicción	Bandera MPM	Índice MPM	Cadena de bin de índice MPM
Modos de MPM (5)	1	0	1
		1	01
		2	001
		3	0001
		4	0000
	0		Código binario truncado (TB)

Antecedentes sobre el modo planar, DC, los modos horizontal, vertical y los modos angulares: Modos de intrapredicción planar (un valor de 0) y DC (un valor de 1). Otros modos de intrapredicción se denominan modos de predicción angular.

En VTM4.0, el siguiente diseño de sintaxis, como se muestra en la Tabla 6, se utiliza para señalar la información del modo de intrapredicción.

Tabla 6 Análisis de sintaxis en VTM4.0

if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]){	
intra_luma_mpm_idx [x0][y0]	ae(v)
}	
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)

intra_luma_ref_idx [x0][y0] puede tener tres valores, 0, 1 o 2, indica qué línea de referencia se utiliza. Cuando **intra_luma_ref_idx** [x0][y0] no está presente, se infiere que es igual a 0.

5 **intra_subpartitions_mode_flag** [x0] [y0] igual a 1 especifica que la unidad de intracodificación actual está particionada aún más (es decir, el modo ISP está habilitado). **intra_subpartitions_mode_flag** [x0][y0] igual a 0 especifica que la unidad de intracodificación actual no está dividida en subparticiones de bloques de transformación rectangulares. Cuando **intra_subpartitions_mode_flag** [x0][y0] no está presente, se infiere que es igual a 0.

10 Los elementos de sintaxis **intra_luma_mpm_flag** [x0][y0], **intra_luma_mpm_idx** [x0][y0] and **intra_luma_mpm_remainder** [x0][y0] especifica el modo de intrapredicción para muestras de luma.

intra_luma_mpm_flag[x0][y0] indica si el modo de intrapredicción del bloque actual se deriva de la lista MPM o no. Si el valor de la **intra_luma_mpm_flag**[x0][y0] es 1, indica que el modo de intrapredicción del bloque actual se deriva de la lista MPM; Si el valor de la **intra_luma_mpm_flag**[x0][y0] es 0, indica que el modo de intrapredicción del bloque actual no se deriva de la lista MPM. Cuando **intra_luma_mpm_flag** [x0][y0] no está presente, se infiere que es igual a 1. **intra_luma_mpm_flag** [x0] [y0] corresponde a **Bandera MPM (MPM flag)** en la Tabla 3.

20 **intra_luma_mpm_idx** corresponde al **índice MPM** en la Tabla 3 y usa código unario truncado como se muestra en la Tabla 3.

intra_luma_mpm_remainder representa código binario truncado del modo que no está en la lista MPM.

25 En una realización 1, se describen un elemento de sintaxis **intra_luma_planar_flag** y ctx.

Según una realización que no forma parte de la invención, se introduce una bandera **intra_luma_planar_flag**.

30 Si un valor de **intra_luma_mpm_flag** es verdadero (por ejemplo, el valor es 1), el decodificador analiza un **intra_luma_planar_flag** para determinar si un modo de intrapredicción de un bloque actual es un modo planar o no. Cuando un valor de **intra_luma_planar_flag** es 1, el modo de intrapredicción del bloque actual es un modo planar. Cuando un valor de **intra_luma_planar_flag** es 0, el modo de intrapredicción del bloque actual NO es un modo planar. Si **intra_luma_planar_flag** no está presente (por ejemplo, esta bandera no se puede analizar desde un flujo de bits en caso de **intra_luma_mpm_flag es igual a 0**), un valor de **intra_luma_planar_flag** se infiere como 0.

35 Si está escrito en un estilo de especificación de borrador de trabajo de VVC, el modo de intrapredicción del bloque actual **IntraPredModeY**[xCb][yCb] se obtiene aplicando el siguiente procedimiento:

- Si **intra_luma_mpm_flag** [xCb][yCb] es igual a 1,

40 - Si **intra_luma_planar_flag**[xCb][yCb] es igual a 1, **IntraPredModeY**[xCb][yCb] se establece igual a **INTRA_PLANAR**.

- De lo contrario, **IntraPredModeY**[xCb][yCb] se establece igual a **candModeList**[**intra_luma_mpm_idx** [xCb][yCb]].

45 Si el modo de intrapredicción no es un modo planar (es decir, un valor de **intra_luma_planar_flag** es 0), el decodificador analiza además un elemento de sintaxis **intra_luma_mpm_idx**.

En un ejemplo, un valor de **intra_luma_planar_flag** está codificado en contexto CABAC (es decir, no codificado en derivación CABAC), y el decodificador analiza la bandera usando el contexto basándose en si el bloque actual es un intra bloque normal o no.

50 En un ejemplo, un valor de **intra_luma_planar_flag** está codificado en contexto CABAC (es decir, no codificado en derivación CABAC), y el decodificador analiza la bandera usando el contexto en función de si el bloque actual es un intra bloque con línea de referencia múltiple (MRL) habilitada o no.

55 En un ejemplo, un valor de **intra_luma_planar_flag** está codificado en contexto CABAC (es decir, no codificado en derivación CABAC), y el decodificador analiza la bandera usando el contexto en función de si el bloque actual es un intra bloque con el modo de codificación de intra subpartición (ISP) habilitado o no.

En un ejemplo, la posición de `intra_luma_planar_flag` se coloca como en la Tabla 7.

Tabla 7: sintaxis con `intra_luma_planar_flag`

if(<code>intra_luma_ref_idx[x0][y0]</code> == 0 && <code>intra_subpartitions_mode_flag[x0][y0]</code> == 0)	
<code>intra_luma_mpm_flag[x0][y0]</code>	ae(v)
if(<code>intra_luma_mpm_flag[x0][y0]</code>){	
<code>intra_luma_planar_flag [x0][y0]</code>	ae(v)
if (! <code>intra_luma_planar_flag [x0][y0]</code>)	
<code>intra_luma_mpm_idx[x0][y0]</code>	ae(v)
}	
else	
<code>intra_luma_mpm_remainder[x0][y0]</code>	ae(v)

Realización 2: `intra_luma_not_planar_flag` y `ctx`

Según un ejemplo, se introduce un elemento de sintaxis `intra_luma_not_planar_flag`.

Si un valor de `intra_luma_mpm_flag` es verdadero (por ejemplo, el valor es 1), el decodificador analiza `intra_luma_not_planar_flag` para determinar si el modo de intrapredicción del bloque actual es un modo planar o no. Cuando un valor de `intra_luma_not_planar_flag` es 1, el modo de intrapredicción del bloque actual NO es un modo planar; Cuando un valor de `intra_luma_not_planar_flag` es 0, el modo de intrapredicción del bloque actual es un modo planar. Si `intra_luma_not_planar_flag` no está presente en un flujo de bits, un valor de `intra_luma_not_planar_flag` se infiere como 1. En un ejemplo, la posición de `intra_luma_not_planar_flag` se coloca como en la Tabla 8.

Si un modo de intrapredicción de un bloque actual no es un modo planar (es decir, un valor de `intra_luma_not_planar_flag` es 1), el decodificador analiza además un elemento de sintaxis `intra_luma_mpm_idx`.

Escrito en un estilo de especificación de borrador de trabajo de VVC, el modo de intrapredicción del bloque actual `IntraPredModeY[xCb][yCb]` se obtiene aplicando el siguiente procedimiento:

- Si `intra_luma_mpm_flag[ xCb ][ yCb ]` es igual a 1,

- Si `intra_luma_not_planar_flag[xCb][yCb]` es igual a 0, `intraPredModeY[xCb][yCb]` se establece igual a `INTRA_PLANAR`.

- De lo contrario, `IntraPredModeY[xCb][yCb]` se establece igual a `candModeList[intra_luma_mpm_idx [xCb][yCb]]`.

En un ejemplo, un valor de `intra_luma_not_planar_flag` está codificado en contexto CABAC (es decir, no codificado en derivación CABAC), y el decodificador analiza la bandera usando el contexto basándose en si el bloque actual es un intra bloque normal o no.

En un ejemplo, un valor de `intra_luma_not_planar_flag` está codificado en contexto CABAC (es decir, no codificado en derivación CABAC), y el decodificador analiza la bandera usando el contexto en función de si el bloque actual es un intra bloque con línea de referencia múltiple (MRL) habilitada o no.

En un ejemplo, un valor de `intra_luma_not_planar_flag` está codificado en contexto CABAC (es decir, no codificado en derivación CABAC), y el decodificador analiza la bandera usando el contexto en función de si el bloque actual es un intra bloque con modo de codificación de intra subpartición (ISP) habilitado o no.

Tabla 8: sintaxis con `intra_luma_not_planar_flag`

if(<code>intra_luma_ref_idx[x0][y0]</code> == 0 && <code>intra_subpartitions_mode_flag[x0][y0]</code> == 0)	
<code>intra_luma_mpm_flag[x0][y0]</code>	ae(v)
if(<code>intra_luma_mpm_flag[x0][y0]</code>){	
<code>intra_luma_not_planar_flag [x0][y0]</code>	ae(v)
if (<code>intra_luma_not_planar_flag [x0][y0]</code>)	
<code>intra_luma_mpm_idx[x0][y0]</code>	ae(v)
}	
else	
<code>intra_luma_mpm_remainder[x0][y0]</code>	ae(v)

Realización 3: **intra_luma_planar_flag** y MRL, y decodificación de derivación

Según una forma de realización que no forma parte de la invención, una bandera **intra_luma_planar_flag** se introduce.

Si un valor de **intra_luma_mpm_flag** es verdadero (por ejemplo, el valor es 1), el decodificador analiza si un valor de la línea de índice de referencia (**intra_luma_ref_idx**) es 0 o no. Si un valor de la línea de índice de referencia es 0, un valor de la **intra_luma_planar_flag** se analiza para determinar si el modo de intrapredicción del bloque actual es un modo predicción planar o no. Cuando un valor del **intra_luma_planar_flag** es 1, el modo de intrapredicción del bloque actual es un modo planar. Cuando un valor del **intra_luma_planar_flag** es 0, el modo de intrapredicción del bloque actual NO es un modo planar. Si **intra_luma_planar_flag** no está presente, un valor de la **intra_luma_planar_flag** se infiere como 0.

Si un modo de intrapredicción de un bloque actual no es un modo planar (es decir, un valor de la **intra_luma_planar_flag** es 0), el decodificador analiza además **intra_luma_mpm_idx**.

En un ejemplo, un valor de la **intra_luma_planar_flag** es **codificada con derivación CABAC** (es decir, no codificado en contexto CABAC), y el decodificador analiza la bandera sin utilizar ningún contexto CABAC.

La posición de **intra_luma_planar_flag** se coloca como en la Tabla 9.

Tabla 9: sintaxis con **intra_luma_planar_flag** y está codificado con derivación

if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_planar_flag [x0][y0]	ae(v)
if (!intra_luma_planar_flag [x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
}	
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)

Realización 4: **intra_luma_not_planar_flag** y MRL, y decodificación de derivación

Según una realización de la invención, una bandera **intra_luma_not_planar_flag** es introducida.

Si un valor de **intra_luma_mpm_flag** es verdadero (el valor es 1), el decodificador analiza si un valor de la línea de índice de referencia es 0 o no. Si un valor de la línea de índice de referencia es 0, **intra_luma_not_planar_flag** se analiza para determinar si el modo de intrapredicción del bloque actual es un modo planar o no. Cuando un valor del **intra_luma_not_planar_flag** es 1, el modo de intrapredicción del bloque actual NO es un modo planar. Cuando un valor del **intra_luma_planar_flag** es 0, el modo de intrapredicción del bloque actual es un modo planar. Si **intra_luma_not_planar_flag** no está presente, un valor de la **intra_luma_not_planar_flag** se infiere como 1.

Si un modo de intrapredicción de un bloque actual no es un modo planar (es decir, un valor de la **intra_luma_not_planar_flag** es 1), el decodificador analiza además **intra_luma_mpm_idx**.

En un ejemplo, un valor de la **intra_luma_not_planar_flag** es **codificado con derivación CABAC** (es decir, no codificado en contexto CABAC), y el decodificador analiza la bandera sin utilizar ningún contexto CABAC.

La posición de **intra_not_luma_planar_flag** se coloca como en la Tabla 10.

Tabla 10: sintaxis con **intra_luma_not_planar_flag** y está codificado con derivación

if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_not_planar_flag [x0][y0]	ae(v)
if (intra_luma_not_planar_flag [x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
}	

else	
intra_luma_mpm_remainder [x0][y0]	ae(v)

Realización 5: **intra_luma_not_planar_flag** y MRL, y codificado en contexto cabac (la diferencia con la realización 4 es que **intra_luma_not_planar_flag** está codificado en contexto cabac en lugar de codificado con derivación).

Según una realización de la invención, una bandera **intra_luma_not_planar_flag** es introducida.

El decodificador analiza un valor que indica el índice de línea de referencia utilizado. Cuando el valor de la línea de índice de referencia es 0, el decodificador analiza un valor de **intra_luma_mpm_flag**.

Si un valor de **intra_luma_mpm_flag** es verdadero (el valor es 1), el decodificador verifica si un valor de la línea de índice de referencia es 0 o no. Si un valor de la línea de índice de referencia es 0, **intra_luma_not_planar_flag** se analiza para determinar si el modo de intrapredicción del bloque actual es un modo planar o no. Cuando un valor del **intra_luma_not_planar_flag** es 1, el modo de intrapredicción del bloque actual NO es un modo planar. Cuando un valor del **intra_luma_not_planar_flag** es 0, el modo de intrapredicción del bloque actual es un modo planar. Si **intra_luma_not_planar_flag** no está presente, un valor de la **intra_luma_not_planar_flag** se infiere como 1.

Si un modo de intrapredicción de un bloque actual no es un modo planar (es decir, un valor de la **intra_luma_not_planar_flag** es 1), el decodificador analiza además **intra_luma_mpm_idx**.

En un ejemplo, un valor de **intra_luma_not_planar_flag** está codificado en contexto CABAC (es decir, no codificado en derivación CABAC), y el decodificador analiza la bandera usando el contexto basándose en si el bloque actual es un intra bloque normal o no.

En un ejemplo, un valor de **intra_luma_not_planar_flag** está codificado en contexto CABAC (es decir, no codificado en derivación CABAC), y el decodificador analiza la bandera usando el contexto en función de si el bloque actual es un intra bloque con línea de referencia múltiple (MRL) habilitada o no.

En un ejemplo, un valor de **intra_luma_not_planar_flag** está codificado en contexto CABAC (es decir, no codificado en derivación CABAC), y el decodificador analiza la bandera usando el contexto en función de si el bloque actual es un intra bloque con modo de codificación de intra subpartición (ISP) habilitado o no.

La posición de **intra_not_luma_planar_flag** se coloca como en la Tabla 11.

Tabla 11: sintaxis con **intra_luma_not_planar_flag** y es codificado con **contexto cabac**

if(intra_luma_ref_idx [x0][y0] == 0 && intra_subpartitions_mode_flag [x0][y0] == 0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag [x0][y0]){	
if(intra_luma_ref_idx [x0][y0] == 0)	
intra_luma_not_planar_flag [x0][y0]	ae(v)
if (intra_luma_not_planar_flag [x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
}	
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)

En un ejemplo,

Add **intra_luma_not_planar_flag**

if(intra_luma_mpm_flag [x0][y0]) {	
if(intra_luma_ref_idx [x0][y0] == 0)	
intra_luma_not_planar_flag [x0][y0]	ae(v)
if(intra_luma_not_planar_flag [x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
} else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	

intra_luma_mpm_flag[][]	FL	cMax = 1
intra_luma_not_planar_flag[][]	FL	cMax = 1
intra_luma_mpm_idx[][]	TR	cMax = 4, cRiceParam = 0

Cuando intra_luma_not_planar_flag[x0][y0] no está presente, se infiere que es igual a 1.

La siguiente Tabla 12 muestra un ejemplo de asignación de ctxInc a elementos de sintaxis con bins codificados por contexto.

Tabla 12 - Asignación de ctxInc a elementos de sintaxis con bins codificados por contexto

Elemento de sintaxis	binIdx					
	0	1	2	3	4	>= 5
intra_subpartition_split_flag	0	na	na	na	na	na
intra_luma_mpm_flag[][]	0	na	na	na	na	na
intra_luma_not_planar_flag[][]	intra_subpartitions_mode_flag	na	na	na	na	na
intra_luma_mpm_idx[][]	bypass (derivación)	bypass (derivación)	bypass (derivación)	bypass (derivación)	na	na
intra_luma_mpm_remainder[][]	bypass (derivación)	bypass (derivación)	bypass (derivación)	bypass (derivación)	bypass (derivación)	bypass (derivación)

En una realización, se construye una lista de MPM de acuerdo con el siguiente proceso.

8.4.2 Proceso de derivación para el modo de intrapredicción de luma

Las entradas a este proceso son:

- una ubicación de luma (xCb, yCb) que especifica la muestra superior izquierda del bloque de codificación de luma actual en relación con la muestra de luma superior-izquierda de la imagen actual,
- una variable cbWidth que especifica el ancho del bloque de codificación actual en muestras de luma,
- una variable cbHeight que especifica la altura del bloque de codificación actual en muestras de luma.

En este proceso, se deriva el modo de intrapredicción de luma IntraPredModeY[xCb][yCb].

La Tabla 13 especifica el valor para el modo de intrapredicción IntraPredModeY[xCb][yCb] y los nombres asociados.

Tabla 13 - Especificación del modo de intrapredicción y nombres asociados

Modo de intrapredicción	Nombre asociado
0	INTRA_PLANAR
1	INTRA_DC
2..66	INTRA_ANGULAR2..INTRA_ANGULAR66
81..83	INTRA_LT_CCLM, INTRA_L_CCLM, INTRA_T_CCLM

NOTA - : Los modos de intrapredicción INTRA_LT_CCLM, INTRA_L_CCLM e

INTRA_T_CCLM solo se aplican a componentes de croma.

IntraPredModeY[xCb][yCb] se deriva de la siguiente manera:

- Si intra_luma_not_planar_flag[xCb][yCb] es igual a 1, se aplican los siguientes pasos ordenados:

1. Las ubicaciones vecinas (xNbA, yNbA) y (xNbB, yNbB) se establecen en (xCb - 1, yCb + cbHeight - 1) y (xCb + cbWidth - 1, yCb - 1), respectivamente.

2. Para que X sea reemplazado por A o B, las variables candIntraPredModeX se derivan de la siguiente manera:

- Se invoca el proceso de derivación de disponibilidad para un bloque: El proceso de verificación de disponibilidad de bloques vecinos se invoca con la ubicación (xCurr, yCurr) establecida en (xCb, yCb) y la ubicación vecina (xNbY, yNbY) establecida en (xNbX, yNbX) como entradas, y la salida se asigna a availableX.

5 - El modo de intrapredicción candidato candIntraPredModeX se deriva de la siguiente manera:

- Si una o más de las siguientes condiciones son verdaderas, candIntraPredModeX se establece igual a INTRA_PLANAR.

10 - La variable availableX es igual a FALSO.

- CuPredMode[xNbX][yNbX] no es igual a MODE_INTRA y ciip_flag[xNbX][yNbX] no es igual a 1.

- pcm_flag[xNbX][yNbX] es igual a 1.

15 - X es igual a B y yCb - 1 es menor que ((yCb >> CtbLog2SizeY) << CtbLog2SizeY).

- De lo contrario, candIntraPredModeX se establece igual a IntraPredModeY[xNbX][yNbX].

20 donde X se reemplaza ya sea por el bloque vecino A o B, una ubicación de luma (xCb, yCb) que especifica la muestra superior izquierda del bloque de codificación de luma actual en relación con la muestra de luma superior-izquierda de la imagen actual, la ubicación (xNbA, yNbA) del vecino A se establece igual a (xCb - 1, yCb + cbHeight - 1), la ubicación (xNbB, yNbB) del vecino B se establece igual a (xCb + cbWidth - 1, yCb - 1), una variable cbWidth que especifica el ancho del bloque de codificación actual en muestras de luma, una variable cbHeight que especifica la altura del bloque de codificación actual en muestras de luma, CuPredMode representa un modo de predicción actual, SizeY representa el tamaño de los componentes Y de un bloque de árbol de codificación.

3. CandModeList[x] con x = 0..4 se deriva de la siguiente manera:

30 - Si candIntraPredModeB es igual a candIntraPredModeA y candIntraPredModeA es mayor que INTRA_DC, candModeList[x] con x = 0..4 se deriva de la siguiente manera:

$$\text{candModeList}[0] = \text{candIntraPredModeA} \quad (8-10)$$

35 $\text{candModeList}[1] = 2 + ((\text{candIntraPredModeA} + 61) \% 64) \quad (8-12)$

$$\text{candModeList}[2] = 2 + ((\text{candIntraPredModeA} - 1) \% 64) \quad (8-13)$$

40 $\text{candModeList}[3] = \text{INTRA_DC} \quad (8-11)$

$$\text{candModeList}[4] = 2 + ((\text{candIntraPredModeA} + 60) \% 64) \quad (8-14)$$

- De lo contrario, si candIntraPredModeB no es igual a candIntraPredModeA y candIntraPredModeA o candIntraPredModeB es mayor que INTRA_DC, se aplica lo siguiente:

45 - Las variables minAB y maxAB se derivan como sigue:

$$\text{minAB} = \text{Min}(\text{candIntraPredModeA}, \text{candIntraPredModeB}) \quad (8-24)$$

50 $\text{maxAB} = \text{Max}(\text{candIntraPredModeA}, \text{candIntraPredModeB}) \quad (8-25)$

- Si candIntraPredModeA y candIntraPredModeB son mayores que INTRA_DC, candModeList[x] con x = 0..4 se deriva de la siguiente manera:

55 $\text{candModeList}[0] = \text{candIntraPredModeA} \quad (8-27)$

$$\text{candModeList}[1] = \text{candIntraPredModeB} \quad (8-29)$$

60 $\text{candModeList}[2] = \text{INTRA_DC} \quad (8-29)$

- Si maxAB - minAB está en el rango de 2 a 62, inclusive, se aplica lo siguiente:

$$\text{candModeList}[3] = 2 + ((\text{maxAB} + 61) \% 64) \quad (8-30)$$

65 $\text{candModeList}[4] = 2 + ((\text{maxAB} - 1) \% 64) \quad (8-31)$

– De lo contrario, se aplica lo siguiente:

$$\text{candModeList}[3] = 2 + ((\text{maxAB} + 60) \% 64) \quad (8-32)$$

$$5 \quad \text{candModeList}[4] = 2 + ((\text{maxAB}) \% 64) \quad (8-33)$$

– De lo contrario ($\text{candIntraPredModeA}$ o $\text{candIntraPredModeB}$ es mayor que INTRA_DC), $\text{candModeList}[x]$ con $x = 0..4$ se deriva de la siguiente manera:

$$10 \quad \text{candModeList}[0] = \text{maxAB} \quad (8-65)$$

$$\text{candModeList}[1] = \text{INTRA_D} \quad (8-66)$$

$$\text{candModeList}[2] = 2 + ((\text{maxAB} + 61) \% 64) \quad (8-66)$$

$$15 \quad \text{candModeList}[3] = 2 + ((\text{maxAB} - 1) \% 64) \quad (8-67)$$

$$\text{candModeList}[4] = 2 + ((\text{maxAB} + 60) \% 64) \quad (8-68)$$

20 – De lo contrario, se aplica lo siguiente:

$$\text{candModeList}[0] = \text{INTRA_DC} \quad (8-71)$$

$$\text{candModeList}[1] = \text{INTRA_ANGULAR50} \quad (8-72)$$

$$25 \quad \text{candModeList}[2] = \text{INTRA_ANGULAR18} \quad (8-73)$$

$$\text{candModeList}[3] = \text{INTRA_ANGULAR46} \quad (8-74)$$

$$30 \quad \text{candModeList}[4] = \text{INTRA_ANGULAR54} \quad (8-75)$$

4. $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se obtiene aplicando el siguiente procedimiento:

35 – Si $\text{intra_luma_mpm_flag}[\text{xCb}][\text{yCb}]$ es igual a 1, $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se establece igual a $\text{candModeList}[\text{intra_luma_mpm_idx}[\text{xCb}][\text{yCb}]]$.

– De lo contrario, $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se deriva aplicando los siguientes pasos ordenados:

40 1. Cuando $\text{candModeList}[i]$ es mayor que $\text{candModeList}[j]$ para $i = 0..3$ y para cada $i, j = (i + 1) .. 4$, ambos valores se intercambian de la siguiente manera:

$$(\text{candModeList}[i], \text{candModeList}[j]) = \text{Swap}(\text{candModeList}[i], \text{candModeList}[j]) \quad (8-94)$$

45 2. $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se deriva mediante los siguientes pasos ordenados:

i. $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se establece igual a $\text{intra_luma_mpm_remainder}[\text{xCb}][\text{yCb}]$.

ii. El valor de $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se incrementa en uno.

50 iii. Para i igual a 0 a 4, inclusive, cuando $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ es mayor o igual que $\text{candModeList}[i]$, el valor de $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se incrementa en uno.

– De lo contrario ($\text{intra_luma_not_planar_flag}[\text{xCb}][\text{yCb}]$ es igual a 0), $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se establece igual a INTRA_PLANAR .

55 La variable $\text{IntraPredModeY}[x][y]$ con $x = \text{xCb}.. \text{xCb} + \text{cbWidth} - 1$ e $y = \text{yCb}.. \text{yCb} + \text{cbHeight} - 1$ se establece para que sea igual a $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$.

60 En un ejemplo, el modo de intrapredicción del bloque actual (por ejemplo, el nombre de la variable es Luma_intra_mode) no es un modo planar ni un modo de intrapredicción en una lista de 5-MPM, su valor se deriva y la predicción de este modo de intrapredicción es realizado de la siguiente manera:

1. Derivar el valor de $\text{intra_luma_mpm_remainder}$ del flujo de bits y luma_intra_mode se establece en el valor de $\text{intra_luma_mpm_remainder}$.

65 2. El valor de $\text{IntraPredModeY}[\text{xCb}][\text{yCb}]$ se incrementa en uno.

3. Para i igual a 0 a 4 (comparar con cada entrada en la lista de 5-MPM), cuando `luma_intra_mode` es mayor o igual que `MPM[i]` (aquí `MPM[0]` corresponde a la primera entrada en la lista MPM, `MPM[1]` corresponde a la segunda entrada en la lista MPM, y así sucesivamente), el valor de `luma_intra_mode` se incrementa en uno.

4. Utilizando el modo de intrapredicción derivado (`luma_intra_mode`) y las muestras de referencia correspondientes para la intrapredicción de desempeño.

1 siempre la primera entrada en una lista implícita de 6-MPM. Esta lista implícita de 6-MPM se compone de planar (siempre la primera entrada) y otros 5 modos (que pueden describirse explícitamente como lista de 5-MPM).

Esto se debe a que si el modo planar está siempre en la primera posición de la lista implícita de 6-MPM, se puede dividir de la lista implícita de MPM y una bandera especial (por ejemplo, una bandera `intra_luma_planar_flag` o una bandera `intra_luma_not_planar_flag`) se asigna para indicar si el modo de intrapredicción del bloque actual es planar o no.

En un ejemplo, considerando introducir `intra_luma_not_planar_flag[x0][y0]` como una bandera especial para indicar si el modo planar es igual al modo de intrapredicción del bloque actual. Cuando está en modo planar, `intra_luma_not_planar_flag` se establece en 0, de lo contrario, se establece en 1. En este caso, el primer bin de `mpm_idx` es equivalente a la bandera especial `intra_luma_not_planar_flag`.

Cuando el modo de intrapredicción del bloque actual es planar, el índice MPM es igual a 0 y el intervalo codificado es 0. Cuando el modo de intrapredicción del bloque actual es No planar, el índice MPM igual no es igual a 0, y el primer intervalo del índice MPM siempre es igual a 1.

En un ejemplo, `intra_luma_mpm_idx` se codifica utilizando un código unario truncado (TR),

	Valor	TR
<code>Mpm_idx</code>	0	0
<code>Mpm_idx</code>	1	1 0
<code>Mpm_idx</code>	2	1 1 0
<code>Mpm_idx</code>	3	1 1 1 0
<code>Mpm_idx</code>	4	1 1 1 1.

Según una realización de la invención (**rama no angular**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como `left_mode` y el modo de intrapredicción del bloque vecino superior se denota como `above_mode`.

- Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

- `Left_mode` no es modo angular y `above_mode` no es modo angular

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a `MPM[1]`, la 2ª entrada corresponde a `MPM[2]`, y así sucesivamente):

`MPM[1]`: DC

`MPM[2]`: modo vertical (es decir, 50)

`MPM[3]`: modo horizontal (es decir, 18)

`MPM[4]`: vertical-4 (es decir, 46)

`MPM[5]`: vertical+4 (es decir, 54)

- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

5 Según una realización de la invención (**rama no angular**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

- Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

- Left_mode no es modo angular y above_mode no es modo angular

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

MPM[1]: DC

MPM[2]: modo vertical (es decir, 50)

MPM[3]: modo horizontal (es decir, 18)

MPM[4]: 66

MPM[5]: 2

- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**rama no angular**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

- Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

- Left_mode no es modo angular y above_mode no es modo angular

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

MPM[1]: DC

MPM[2]: modo vertical (es decir, 50)

MPM[3]: modo horizontal (es decir, 18)

MPM[4]: 2

MPM[5]: 34

- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**rama no angular**):

- 5 • Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.
- 10 • Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.
- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,
 - 15 ○ Left_mode no es modo angular y above_mode no es modo angular
 luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):
 - 20 MPM[1]: DC
 - MPM[2]: modo vertical (es decir, 50)
 - MPM[3]: modo horizontal (es decir, 18)
 - 25 MPM[4]: 66
 - MPM[5]: 34
- 30 • Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**una rama angular, una angular la otra no angular**):

- 35 • Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.
- 40 • Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.
- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,
 - 45 ○ Uno de los modos left_mode y above_mode es el modo angular, el otro es el modo no angular.
 luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):
 - 50 definir un modo angular como ang y obtenga lo siguiente:
 - si left_mode es modo angular, ang = modo izquierdo
 - de lo contrario (el modo anterior es el modo angular) ang = el modo superior/anterior
 - 55 MPM[1]: ang
 - MPM[2]: DC
 - 60 $MPM[3]: 2 + ((ang + 61) \% 64)$
 - $MPM[4]: 2 + ((ang - 1) \% 64)$
 - 65 $MPM[5]: 2 + ((ang + 60) \% 64)$

- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

5 Según una realización de la invención (**una rama angular, una angular la otra no angular**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

- Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

- Uno de los modos left_mode y above_mode es el modo angular, el otro es el modo no angular.

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

definir un modo angular como ang y obtenga lo siguiente:

si left_mode es modo angular, ang = modo izquierdo

de lo contrario (el modo anterior es el modo angular) ang = el modo superior/anterior

MPM[1]: DC

MPM[2]: ang

MPM[3]: $2 + ((ang + 61) \% 64)$

MPM[4]: $2 + ((ang - 1) \% 64)$

MPM[5]: $2 + ((ang + 60) \% 64)$

- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**una rama angular, una angular la otra no angular**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

- Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

- Uno de los modos left_mode y above_mode es el modo angular, el otro es el modo no angular.

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

definir un modo angular como ang y obtenga lo siguiente:

si left_mode es modo angular, ang = modo izquierdo

de lo contrario (el modo anterior es el modo angular) ang = el modo superior/anterior

MPM[1]: ang

MPM[2]: $2 + ((ang + 61) \% 64)$

5 MPM[3]: DC

MPM[4]: $2 + ((ang - 1) \% 64)$

MPM[5]: $2 + ((ang + 60) \% 64)$

10 • Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**una rama angular, una angular la otra no angular**):

15 • Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

• Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

25 • Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

○ Uno de los modos left_mode y above_mode es el modo angular, el otro es el modo no angular.

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

30 definir un modo angular como ang y obtenga lo siguiente:

definir un modo angular como ang y obtenga lo siguiente:

si left_mode es modo angular, ang = modo izquierdo

35 de lo contrario (el modo anterior es el modo angular) ang = el modo superior/anterior

MPM[0]: Planar

40 MPM[1]: ang

MPM[2]: $2 + ((ang + 61) \% 64)$

MPM[3]: $2 + ((ang - 1) \% 64)$

45 MPM[4]: DC

MPM[5]: $2 + ((ang + 60) \% 64)$

50 • Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**una rama angular, una angular la otra no angular**):

55 • Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

60 • Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

• Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

65 ○ Uno de los modos left_mode y above_mode es el modo angular, el otro es el modo no angular.

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

- 5 definir un modo angular como ang y obtenga lo siguiente:
- si left_mode es modo angular, $ang = \text{modo izquierdo}$
- 10 de lo contrario (el modo anterior es el modo angular) $ang = \text{el modo superior/anterior}$
- MPM[1]: ang
- MPM[2]: $2 + ((ang + 61) \% 64)$
- 15 MPM[3]: $2 + ((ang - 1) \% 64)$
- MPM[4]: $2 + ((ang + 60) \% 64)$
- 20 MPM[5]: DC
- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

25 Según una realización de la invención (**una rama angular, dos son iguales angulares**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.
- Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.
- 35 • Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,
 - left_mode y above_mode son ambos modos angulares y son iguales.

40 luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

- MPM[1]: modo izquierdo
- 45 MPM[2]: $2 + ((\text{modo izquierdo} + 61) \% 64)$
- MPM[3]: $2 + ((\text{modo izquierdo} - 1) \% 64)$
- MPM[4]: DC
- 50 MPM[5]: $2 + ((\text{modo izquierdo} + 60) \% 64)$
- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

55 Según una realización de la invención (**una rama angular, dos son iguales angulares**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.
- Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

65

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,
 - left_mode y above_mode son ambos modos angulares y son iguales.

5 luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

MPM[1]: DC

10 MPM[2]: modo izquierdo

MPM[3]: $2 + ((\text{modo izquierdo} + 61) \% 64)$

15 MPM[4]: $2 + ((\text{modo izquierdo} - 1) \% 64)$

MPM[5]: $2 + ((\text{modo izquierdo} + 60) \% 64)$

- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**una rama angular, dos son iguales angulares**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

- Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,
 - left_mode y above_mode son ambos modos angulares y son iguales.

35 luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

40 MPM[1]: modo izquierdo

MPM[2]: DC

MPM[3]: $2 + ((\text{modo izquierdo} + 61) \% 64)$

45 MPM[4]: $2 + ((\text{modo izquierdo} - 1) \% 64)$

MPM[5]: $2 + ((\text{modo izquierdo} + 60) \% 64)$

- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**una rama angular, dos son iguales angulares**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

- Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,
 - left_mode y above_mode son ambos modos angulares y son iguales.

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

MPM[1]: modo izquierdo

MPM[2]: $2 + ((\text{modo izquierdo} + 61) \% 64)$

MPM[3]: DC

MPM[4]: $2 + ((\text{modo izquierdo} - 1) \% 64)$

MPM[5]: $2 + ((\text{modo izquierdo} + 60) \% 64)$

- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**una rama angular, dos son iguales angulares**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

- Paso 2: determinar si los modos de intrapredicción de los bloques vecinos son modos angulares o no.

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

- left_mode y above_mode son ambos modos angulares y son iguales.

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

MPM[1]: modo izquierdo

MPM[2]: $2 + ((\text{modo izquierdo} + 61) \% 64)$

MPM[3]: $2 + ((\text{modo izquierdo} - 1) \% 64)$

MPM[4]: $2 + ((\text{modo izquierdo} + 60) \% 64)$

MPM[5]: DC

- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

•

Según una realización de la invención (**rama de dos ángulos**):

- Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

- Paso 2: determinar si el modo de intrapredicción de los bloques vecinos son modos angulares o no.

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

- left_mode y above_mode son dos modos angulares diferentes.

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

5 MPM[1]: left_mode

MPM[2]: above_mode

MPM[3]: DC

10

– Las variables minAB y maxAB se derivan como sigue:

minAB = Min(above_mode, left_mode)

15 maxAB = Max(above_mode, left_mode)

– Si maxAB - minAB está en el rango de 2 a 62, inclusive, se aplica lo siguiente:

MPM [4] = 2 + ((maxAB + 61) % 64) (8-30)

20

MPM [5] = 2 + ((maxAB - 1) % 64) (8-31)

– De lo contrario, se aplica lo siguiente:

25 MPM [4] = 2 + ((maxAB + 60) % 64) (8-32)

MPM [5] = 2 + ((maxAB) % 64) (8-33)

30 • Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**rama de dos ángulos**):

35 • Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

40 • Paso 2: determinar si el modo de intrapredicción de los bloques vecinos son modos angulares o no.

• Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

45 ○ left_mode y above_mode son dos modos angulares diferentes.

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

50 MPM[1]: above_mode

MPM[2]: left_mode

MPM[3]: DC

55

– Las variables minAB y maxAB se derivan como sigue:

minAB = Min(above_mode, left_mode)

60 maxAB = Max(above_mode, left_mode)

– Si maxAB - minAB está en el rango de 2 a 62, inclusive, se aplica lo siguiente:

MPM [4] = 2 + ((maxAB + 61) % 64) (8-30)

65

MPM [5] = 2 + ((maxAB - 1) % 64) (8-31)

– De lo contrario, se aplica lo siguiente:

$$\text{MPM}[4] = 2 + ((\text{maxAB} + 60) \% 64) \quad (8-32)$$

$$\text{MPM}[5] = 2 + ((\text{maxAB}) \% 64) \quad (8-33)$$

• Paso 6: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**rama de dos ángulos**):

• Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

• Paso 2: determinar si el modo de intrapredicción de los bloques vecinos son modos angulares o no.

• Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,

◦ left_mode y above_mode son dos modos angulares diferentes.

luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

MPM[1]: left_mode

MPM[2]: DC

MPM[3]: above_mode

– Las variables minAB y maxAB se derivan como sigue:

minAB = Min(above_mode, left_mode)

maxAB = Max(above_mode, left_mode)

– Si maxAB - minAB está en el rango de 2 a 62, inclusive, se aplica lo siguiente:

$$\text{MPM}[4] = 2 + ((\text{maxAB} + 61) \% 64) \quad (8-30)$$

$$\text{MPM}[5] = 2 + ((\text{maxAB} - 1) \% 64) \quad (8-31)$$

– De lo contrario, se aplica lo siguiente:

$$\text{MPM}[4] = 2 + ((\text{maxAB} + 60) \% 64) \quad (8-32)$$

$$\text{MPM}[5] = 2 + ((\text{maxAB}) \% 64) \quad (8-33)$$

• Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

Según una realización de la invención (**rama de dos ángulos**):

• Paso 1: obtener un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino anterior según la FIG. 10. Si un modo de intrapredicción del bloque vecino no está disponible (por ejemplo, el bloque vecino está fuera de los límites de la imagen o el bloque vecino está intercodificado), el modo de intrapredicción del bloque vecino se establece como planar. El modo de intrapredicción del vecino izquierdo se denota como left_mode y el modo de intrapredicción del bloque vecino superior se denota como above_mode.

• Paso 2: determinar si el modo de intrapredicción de los bloques vecinos son modos angulares o no.

- Paso 3: si todas las siguientes condiciones (viñetas) se evalúan como verdaderas,
 - left_mode y above_mode son dos modos angulares diferentes.

5 luego, se construye una lista de modos de intrapredicción que comprende N entradas de modos de intrapredicción como sigue (por ejemplo, N es igual a 5, donde la 1ª entrada corresponde a MPM[1], la 2ª entrada corresponde a MPM[2], y así sucesivamente):

MPM[1]: above_mode

10 MPM[2]: DC

MPM[3]: left_mode

15 – Las variables minAB y maxAB se derivan como sigue:

minAB = Min(above_mode, left_mode)

maxAB = Max(above_mode, left_mode)

20 – Si maxAB - minAB está en el rango de 2 a 62, inclusive, se aplica lo siguiente:

MPM [4] = 2 + ((maxAB + 61) % 64) (8-30)

25 MPM [5] = 2 + ((maxAB - 1) % 64) (8-31)

– De lo contrario, se aplica lo siguiente:

MPM [4] = 2 + ((maxAB + 60) % 64) (8-32)

30 MPM [5] = 2 + ((maxAB) % 64) (8-33)

- Paso 4: predecir el bloque actual según un índice (indicado como índice MPM) y las entradas de la lista construida.

35 En una implementación de la realización, N se establece en 6.

En una implementación de la realización, N se establece en 6. En este caso, dicha lista de modos de intrapredicción incluye las primeras 5 entradas enumeradas anteriormente y no la sexta entrada.

40 En una implementación de la realización, N se establece en 4. En este caso, dicha lista de modos de intra predicción incluye las primeras 4 entradas enumeradas anteriormente y no la quinta entrada y la sexta entrada.

45 En una implementación de la realización, dicha lista de modos de intrapredicción se denomina lista MPM.

En una implementación, dicho índice de lista (índice MPM) se señala en el flujo de bits como un indicador. De manera ventajosa, el índice MPM se puede codificar con el Codificador de entropía adaptable al contexto (CABAC). El índice MPM podría codificarse de acuerdo con varios modelos de probabilidad (en otras palabras, contexto) por CABAC.

50 Por ejemplo, el primer bin de la palabra codificada para mpm_index está codificado en contexto CABAC.

En un ejemplo, su contexto se decide en base a si el intra bloque actual se aplica con múltiples líneas de referencia, ISP o intra bloques normales. En total, se crean tres contextos para el contexto CABAC del primer bin de la palabra codificada para mpm_index.

55 Si una lista MPM contiene tanto el modo planar como el modo DC, la herramienta de codificación de múltiples líneas de referencia utiliza una lista de MPM que excluye el modo planar y DC de la lista MPM construida. En este caso, la codificación del modo de intrapredicción de varias líneas de referencia utiliza una lista de 4 MPM. En un ejemplo, una lista de MPM es {Planar (un valor de 0), DC (un valor de 1), VER (un valor de 50), HOR (un valor de 18), VER-4 (un valor de 46) , VER+4 (un valor de 54)} después de la construcción de la lista MPM, luego la codificación del modo de intrapredicción usa una lista 4-MPM de {VER, HOR, VER-4, VER 4} cuando se habilita una línea de referencia múltiple.

65 Si una lista MPM contiene modo planar y no DC, entonces la herramienta de codificación de línea de referencia múltiple utiliza una lista MPM que excluye el planar de la lista MPM construida. En este caso, la codificación del modo de intrapredicción de múltiples líneas de referencia utiliza una lista de 5 MPM. En un ejemplo, una lista MPM es

{Planar, VER, HOR, 43, 60, 3} después de la construcción de la lista MPM, luego una lista de 5 MPM de {VER, HOR, 43, 60, 3} es usada por la codificación de modo de intrapredicción cuando está habilitada la línea de referencia múltiple.

En otro ejemplo, podría ser necesario modificar la sintaxis de una lista MPM sin modo DC. Si una lista de MPM no contiene DC, probablemente el modo DC se usará con mucha frecuencia en la rama que no es MPM (es decir, la bandera de mpm es falso). En este caso, se introduce una nueva sintaxis, DC_mode. DC_mode es igual a 1 indica que el modo de intrapredicción del bloque actual es el modo DC. DC_mode es igual a 0 indica que el modo de intrapredicción del bloque actual no es el modo DC.

Es decir, la antigua sintaxis de no MPM se cambia de

If (mpm_flag)

...

else

intra_mode (using TB)

a

If (mpm_flag)

...

else

DC_mode

If (DC_mode==0)

intra_mode (using TB)

La sintaxis DC_mode se puede codificar con el Codificador de entropía adaptable al contexto (CABAC). El DC_mode podría codificarse de acuerdo con varios modelos de probabilidad (en otras palabras, contexto) por CABAC. Con la introducción de DC_Mode, el valor máximo para binario truncado es 60, es decir, 67 (número total de modos) - 6 (MPM) - 1 (DC). Si bien sin la introducción de DC_mode, el valor máximo para el binario truncado es 61, es decir, 67 (número de modos totales) - 6 (MPM).

En particular, se proporcionan aquí los siguientes métodos de codificación de predicción de un bloque actual implementado por un dispositivo de decodificación o dispositivo de codificación. El dispositivo de decodificación puede ser el decodificador 30 de video de la FIG. 1A o el decodificador 30 de la FIG. 3. El dispositivo de codificación puede ser el codificador/encodificador 20 de video de la FIG. 1A o el codificador 20 de la FIG. 2.

Según una realización 1200 (véase la FIG. 12A), un método de intrapredicción de un bloque actual implementado por un dispositivo de decodificación incluye los siguientes pasos. Paso 1201, el dispositivo obtiene un valor de una primera información de indicación de un bloque actual, el valor de la primera información de indicación indica si el modo de intrapredicción del bloque actual está comprendido en un conjunto de modos más probables. El conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y el modo planar. La primera información de indicación se indica mediante una primera bandera, por ejemplo, intra_luma_mpm_flag. Como se describió anteriormente, cuando el valor de intra_luma_mpm_flag es igual a 1, el modo de intrapredicción del bloque actual se incluye en el conjunto de modos más probables.

Paso 1202, el dispositivo obtiene un valor de una línea de índice de referencia del bloque actual. El valor de la línea de índice de referencia del bloque actual puede indicarse mediante intra_luma_ref_idx. Cuando el valor de la línea de índice de referencia del bloque actual es igual a 0, es decir, intra_luma_ref_idx[x0][y0] = 0, se utiliza la línea de referencia vecina más cercana al bloque actual. No hay secuencia entre el paso 1201 y 1202.

Paso 1203, el dispositivo determina si el valor de la primera información de indicación indica que el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables.

Paso 1204, el dispositivo determina si el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual. No hay secuencia entre el paso 1203 y 1204.

Paso 1205, el dispositivo obtiene un valor de una segunda información de indicación del bloque actual cuando el valor

de la primera información de indicación indica que el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual, el valor de la segunda información de indicación del bloque actual indica si el modo de intrapredicción del bloque actual es el modo planar o no. El valor de la segunda información de indicación es codificación aritmética binaria adaptable al contexto (CABAC) decodificado por contexto. La segunda información de indicación puede indicarse mediante una segunda bandera, por ejemplo, `intra_luma_not_planar_flag`. El dispositivo analiza la segunda bandera utilizando el contexto en función de si el bloque actual es un intra bloque con el modo de codificación de intra subpartición (ISP) habilitado o no.

Cuando la codificación MRL no está habilitada, es decir, cuando el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual, los dispositivos analizan un flujo de bits para obtener el valor de la segunda información de indicación.

Se encuentra que el modo planar no es eficiente cuando el MRL está habilitado (es decir, el índice de la línea de referencia no es igual a 0) en el método convencional. En la realización de la invención, cuando se habilita la codificación MRL, el valor de la segunda información de indicación se obtiene mediante un valor predeterminado. En esta situación, se excluye el modo planar y su bandera correspondiente (es decir, `intra_luma_not_planar_flag`) no se analiza a partir del flujo de bits, pero se infiere como un valor predeterminado (por ejemplo, un valor de 1). En este caso, se guarda un bit para indicar el valor de la bandera planar (es decir, `intra_luma_not_planar`) cuando MRL está habilitado).

La información detallada de esta realización 1200 se muestra en las realizaciones mencionadas anteriormente.

Según una realización 1210 (véase la FIG. 12B), un método de intrapredicción de un bloque actual implementado por un dispositivo de decodificación incluye los siguientes pasos.

Paso 1211, el dispositivo obtiene un valor de una línea de índice de referencia del bloque actual. El valor de la línea de índice de referencia del bloque actual puede indicarse mediante `intra_luma_ref_idx`. Cuando el valor de la línea de índice de referencia del bloque actual es igual a 0, es decir, `intra_luma_ref_idx[x0][y0] = 0`, se utiliza la línea de referencia vecina más cercana al bloque actual.

Paso 1212, el dispositivo determina si el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual.

Paso 1213, cuando el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual, el dispositivo obtiene un valor de una primera información de indicación de un bloque actual, el valor de la primera información de indicación indica si el modo de predicción del bloque actual se compone de un conjunto de modos más probables, el conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y el modo planar. La primera información de indicación se indica mediante una primera bandera, `intra_luma_mpm_flag`. Como se describió anteriormente, cuando el valor de `intra_luma_mpm_flag` es igual a 1, el modo de intrapredicción del bloque actual se incluye en el conjunto de modos más probables.

Paso 1214, el dispositivo determina si el valor de la primera información de indicación indica que el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables.

Paso 1215, el dispositivo obtiene un valor de una segunda información de indicación del bloque actual cuando el valor de la primera información de indicación indica que el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual, el valor de la segunda información de indicación del bloque actual indica si el modo de intrapredicción del bloque actual es el modo planar o no. El valor de la segunda información de indicación es codificación aritmética binaria adaptable al contexto (CABAC) decodificado por contexto. La segunda información de indicación se indica mediante una segunda bandera, `intra_luma_not_planar_flag`. El dispositivo analiza la segunda bandera utilizando el contexto en función de si el bloque actual es un intra bloque con el modo de codificación de intra subpartición (ISP) habilitado o no.

La información detallada de esta realización 1210 se muestra en las realizaciones mencionadas anteriormente.

Según una realización 1300 (véase la FIG. 13), un método de intrapredicción de un bloque actual implementado por un dispositivo de codificación incluye los siguientes pasos. Paso 1301, el dispositivo determina si un modo de intrapredicción de un bloque actual está comprendido en un conjunto de modos más probables. El conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y el modo planar. Paso 1302, el dispositivo determina si la línea de referencia vecina más cercana al bloque actual se usa para la intrapredicción. No hay secuencia entre 1301 y 1302.

Paso 1303, el dispositivo codifica un flujo de bits, en donde el flujo de bits incluye información para indicar si el modo de intrapredicción del bloque actual es el modo planar cuando el modo de intrapredicción del bloque actual está

comprendido en el conjunto de modos más probables y la línea de referencia vecina más cercana al bloque actual se utiliza para la intrapredicción y envía el flujo de bits codificado a un dispositivo de decodificación.

El flujo de bits incluye una primera bandera que indica si el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y una segunda bandera que indica si el modo de intrapredicción del bloque actual es un modo planar. Como se describió anteriormente, la primera bandera es `intra_luma_mpm_flag`. Cuando el valor de `intra_luma_mpm_flag` es igual a 1, el modo de intrapredicción del bloque actual se incluye en el conjunto de modos más probables.

El valor de la segunda información de indicación es codificación aritmética binaria adaptable al contexto (CABAC) decodificado por contexto. La segunda información de indicación se indica mediante una segunda bandera, `intra_luma_not_planar_flag`. El dispositivo analiza la segunda bandera utilizando el contexto en función de si el bloque actual es un intra bloque con el modo de codificación de intra subpartición (ISP) habilitado o no.

El flujo de bits también puede incluir una indicación para indicar el valor de la línea de índice de referencia del bloque actual. El valor de la línea de índice de referencia del bloque actual se indica mediante `intra_luma_ref_idx`, por ejemplo. Cuando el valor de la línea de índice de referencia del bloque actual es igual a 0, es decir, `intra_luma_ref_idx[x0][y0] = 0`, se utiliza la línea de referencia vecina más cercana al bloque actual.

La información detallada de esta realización 1300 se muestra en las realizaciones mencionadas anteriormente.

La FIG. 14 ilustra realizaciones de un dispositivo de decodificación 1400. El dispositivo de decodificación 1400 puede ser el decodificador 30 de video de la FIG. 1A o el decodificador 30 de la FIG. 3. El dispositivo de decodificación 1400 puede usarse para implementar la realización 1200 y las otras realizaciones descritas anteriormente.

El dispositivo incluye una primera unidad de obtención 1401 y una segunda unidad de obtención 1402. La primera unidad de obtención 1401, configurada para obtener un valor de una línea de índice de referencia de un bloque actual, y un valor de una primera información de indicación del bloque actual, el valor de la primera información de indicación indica si un modo de intrapredicción del bloque actual está comprendido en un conjunto de modos más probables. El conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y el modo planar. La segunda unidad de obtención 1402, configurada para obtener un valor de una segunda información de indicación del bloque actual cuando el valor de la primera información de indicación indica que el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y cuando el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual, en donde el valor de la segunda información de indicación del bloque actual indica si el modo de intrapredicción del bloque actual es el modo planar.

El dispositivo 1400 incluye además una unidad de análisis 1403 (no mostrada en la FIG. 14). La unidad de análisis 1403 está configurada para analizar la segunda bandera usando el contexto en función de si el bloque actual es un intra bloque con el modo de codificación de intra subpartición (ISP) habilitado o no.

La FIG. 15 ilustra realizaciones de un dispositivo de codificación 1500 para intrapredicción. El dispositivo de codificación 1500 puede ser el codificador de video 20 de la FIG. 1A o el codificador 20 de la Figura 2. El dispositivo de codificación 1500 puede usarse para implementar la realización 1300 y las otras realizaciones descritas anteriormente.

El dispositivo 1500 incluye una unidad de determinación 1501 y una unidad de codificación 1502. La unidad de determinación 1501, configurada para determinar si un modo de intrapredicción de un bloque actual está comprendido en un conjunto de modos más probables, y para determinar si la línea de referencia vecina más cercana al bloque actual se usa para la intrapredicción. El conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y el modo planar. La unidad de codificación 1502, configurada para codificar un flujo de bits, en donde el flujo de bits incluye información para indicar si el modo de intrapredicción del bloque actual es el modo planar cuando el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y la línea de referencia vecina más cercana al bloque actual se utiliza para la intrapredicción.

A continuación se ofrece una explicación de las aplicaciones del método de codificación así como del método de decodificación como se muestra en las realizaciones mencionadas anteriormente, y un sistema que las usa.

La FIG. 16 es un diagrama de bloques que muestra un sistema 3100 de suministro de contenido para realizar un servicio de distribución de contenido. Este sistema de suministro de contenido 3100 incluye el dispositivo de captura 3102, el dispositivo terminal 3106 y, opcionalmente, incluye la pantalla 3126. El dispositivo de captura 3102 se comunica con el dispositivo terminal 3106 a través del enlace de comunicación 3104. El enlace de comunicación puede incluir el canal de comunicación 13 descrito anteriormente. El enlace de comunicación 3104 incluye, entre otros, WIFI, Ethernet, Cable, inalámbrico (3G/4G/5G), USB o cualquier tipo de combinación de los mismos, o similares.

El dispositivo de captura 3102 genera datos y puede codificar los datos mediante el método de codificación como se muestra en las realizaciones anteriores. Alternativamente, el dispositivo de captura 3102 puede distribuir los datos a un servidor de transmisión continua (no mostrado en las Figuras), y el servidor codifica los datos y transmite los datos codificados al dispositivo terminal 3106. El dispositivo de captura 3102 incluye, pero no se limita a, cámara, teléfono inteligente o tableta, ordenador u ordenador portátil, sistema de videoconferencia, PDA, dispositivo montado en un vehículo o una combinación de cualquiera de ellos, o similares. Por ejemplo, el dispositivo de captura 3102 puede incluir el dispositivo de origen 12 como se describió anteriormente. Cuando los datos incluyen video, el codificador de video 20 incluido en el dispositivo de captura 3102 puede realmente realizar el procesamiento de codificación de video. Cuando los datos incluyen audio (es decir, voz), un codificador de audio incluido en el dispositivo de captura 3102 puede realizar realmente el procesamiento de codificación de audio. Para algunos escenarios prácticos, el dispositivo de captura 3102 distribuye los datos de audio y video codificados multiplexándolos juntos. Para otros escenarios prácticos, por ejemplo en el sistema de videoconferencia, los datos de audio codificados y los datos de video codificados no se multiplexan. El dispositivo de captura 3102 distribuye los datos de audio codificados y los datos de video codificados al dispositivo terminal 3106 por separado.

En el sistema de suministro de contenido 3100, el dispositivo terminal 310 recibe y reproduce los datos codificados. El dispositivo terminal 3106 podría ser un dispositivo con capacidad de recepción y recuperación de datos, como un teléfono inteligente o tableta 3108, ordenador u ordenador portátil 3110, grabadora de video en red (NVR)/grabadora de video digital (DVR) 3112, TV 3114, decodificador (STB) 3116, sistema de videoconferencia 3118, sistema de videovigilancia 3120, asistente digital personal (PDA) 3122, dispositivo montado en un vehículo 3124, o una combinación de cualquiera de ellos, o similar capaz de decodificar los datos codificados antes mencionados. Por ejemplo, el dispositivo terminal 3106 puede incluir el dispositivo de destino 14 como se describió anteriormente. Cuando los datos codificados incluyen video, se prioriza el decodificador de video 30 incluido en el dispositivo terminal para realizar la decodificación de video. Cuando los datos codificados incluyen audio, se prioriza un decodificador de audio incluido en el dispositivo terminal para realizar el procesamiento de decodificación de audio.

Para un dispositivo terminal con su pantalla, por ejemplo, teléfono inteligente o tableta 3108, ordenador u ordenador portátil 3110, grabadora de video en red (NVR)/grabadora de video digital (DVR) 3112, TV 3114, asistente digital personal (PDA) 3122 o vehículo dispositivo montado 3124, el dispositivo terminal puede alimentar los datos decodificados a su pantalla. Para un dispositivo terminal equipado sin pantalla, por ejemplo, STB 3116, sistema de videoconferencia 3118 o sistema de videovigilancia 3120, se contacta con una pantalla externa 3126 para recibir y mostrar los datos decodificados.

Cuando cada dispositivo de este sistema realiza la codificación o decodificación, se puede usar el dispositivo de codificación de imágenes o el dispositivo de decodificación de imágenes, como se muestra en las realizaciones mencionadas anteriormente.

La FIG. 17 es un diagrama que muestra una estructura de un ejemplo del dispositivo terminal 3106. Después de que el dispositivo terminal 3106 recibe el flujo del dispositivo de captura 3102, la unidad de procedimiento de protocolo 3202 analiza el protocolo de transmisión del flujo. El protocolo incluye, entre otros, Protocolo de transmisión en tiempo real (RTSP), Protocolo de transferencia de hipertexto (HTTP), Protocolo de transmisión en vivo HTTP (HLS), MPEG-DASH, Protocolo de transporte en tiempo real (RTP), Protocolo de mensajería en tiempo real (RTMP), o cualquier tipo de combinación de los mismos, o similares.

Después de que la unidad de procedimiento de protocolo 3202 procesa el flujo, se genera el archivo de flujo. El archivo se envía a una unidad de demultiplexación 3204. La unidad de demultiplexación 3204 puede separar los datos multiplexados en los datos de audio codificados y los datos de video codificados. Como se describió anteriormente, para algunos escenarios prácticos, por ejemplo, en el sistema de videoconferencia, los datos de audio codificados y los datos de video codificados no se multiplexan. En esta situación, los datos codificados se transmiten al decodificador de video 3206 y al decodificador de audio 3208 sin pasar a través de la unidad de demultiplexación 3204.

A través del procesamiento de demultiplexación, se generan el flujo elemental de video (ES), el ES de audio y, opcionalmente, los subtítulos. El decodificador de video 3206, que incluye el decodificador de video 30 como se explica en las realizaciones mencionadas anteriormente, decodifica el ES de video mediante el método de decodificación como se muestra en las realizaciones mencionadas anteriormente para generar un fotograma de video, y alimenta estos datos a la unidad síncrona 3212. El decodificador de audio 3208 decodifica el ES de audio para generar un fotograma de audio y alimenta estos datos a la unidad síncrona 3212. Alternativamente, el fotograma de video puede almacenarse en una memoria intermedia (no mostrada en la FIG. Y) antes de alimentarlo a la unidad síncrona 3212. De manera similar, la trama de audio puede almacenarse en un búfer (no mostrado en la FIG. Y) antes de alimentarlo a la unidad síncrona 3212.

La unidad síncrona 3212 sincroniza el fotograma de video y el fotograma de audio, y suministra el video/audio a una pantalla de video/audio 3214. Por ejemplo, la unidad síncrona 3212 sincroniza la presentación de la información de video y audio. La información puede codificarse en la sintaxis utilizando marcas de tiempo relativas a la presentación de datos de audio y visuales codificados y marcas de tiempo relativas a la entrega del flujo de datos en sí.

Si se incluyen subtítulos en la secuencia, el decodificador de subtítulos 3210 decodifica el subtítulo y lo sincroniza con el cuadro de video y el cuadro de audio, y suministra el video/audio/subtítulo a una pantalla 3216 de video/audio/subtítulos.

La presente invención no se limita al sistema mencionado anteriormente, y el dispositivo de codificación de imágenes o el dispositivo de decodificación de imágenes en las realizaciones mencionadas anteriormente se pueden incorporar en otro sistema, por ejemplo, un sistema de automóvil.

Operadores matemáticos

Los operadores matemáticos utilizados en esta solicitud son similares a los utilizados en el lenguaje de programación C. Sin embargo, los resultados de la división de enteros y las operaciones de desplazamiento aritmético se definen con mayor precisión y se definen operaciones adicionales, como la exponenciación y la división en valores reales. Las convenciones de numeración y conteo generalmente comienzan desde 0, por ejemplo, "el primero" es equivalente al 0-ésimo, "el segundo" es equivalente al 1-ésimo, etc.

Operadores aritméticos

Los siguientes operadores aritméticos se definen de la siguiente manera:

+	Adición/suma
-	Resta (como operador de dos argumentos) o negación (como operador de prefijo unario)
*	Multiplicación, incluida la multiplicación de matrices
x^y	Exponenciación. Especifica x elevado a y. En otros contextos, dicha notación se usa para superíndices que no están destinados a ser interpretados como exponenciación.
/	División de enteros con truncamiento del resultado hacia cero. Por ejemplo, 7/4 y -7/-4 se truncan a 1 y -7/4 y 7/-4 se truncan a -1.
÷	Se usa para denotar división en ecuaciones matemáticas donde no se pretende truncamiento o redondeo.
$\frac{x}{y}$	Se usa para denotar división en ecuaciones matemáticas donde no se pretende truncamiento o redondeo.
$\sum_{i=x}^y f(i)$	La suma de f(i) con i tomando todos los valores enteros desde x hasta y e incluyendo y.
$x \% y$	Módulo. Resto de x dividido por y, definido solo para enteros x e y con $x \geq 0$ e $y > 0$.

Operadores lógicos

Los siguientes operadores lógicos se definen de la siguiente manera:

$x \&\& y$	Booleano lógico "y" de x e y
$x \mid \mid y$	Booleano lógico "o" de x e y
!	"No" lógico booleano
$x ? y : z$	Si x es VERDADERO o no es igual a 0, se evalúa como el valor de y; de lo contrario, se evalúa al valor de z.

Operadores relacionales

Los siguientes operadores relacionales se definen de la siguiente manera:

>	Mayor que
>=	Mayor o igual que
<	Menos de
<=	Menor o igual a
==	Igual a

!= No es igual a

Cuando se aplica un operador relacional a un elemento sintáctico o variable al que se le ha asignado el valor "na" (no aplicable), el valor "na" se trata como un valor distinto para el elemento sintáctico o variable. Se considera que el valor "na" no es igual a ningún otro valor.

Operadores de bit a bit

Los siguientes operadores bit a bit se definen como sigue:

& Bit a bit "y". Cuando se opera con argumentos enteros, opera con una representación en complemento a dos del valor entero. Cuando se opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se amplía agregando más bits significativos iguales a 0.

| Bit a bit "or". Cuando se opera con argumentos enteros, opera con una representación en complemento a dos del valor entero. Cuando se opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se amplía agregando más bits significativos iguales a 0.

^ Bit a bit "o exclusivo". Cuando se opera con argumentos enteros, opera con una representación en complemento a dos del valor entero. Cuando se opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se amplía agregando más bits significativos iguales a 0.

x >> y Desplazamiento aritmético a la derecha de una representación entera en complemento a dos de x por y dígitos binarios. Esta función está definida solo para valores enteros no negativos de y. Los bits desplazados a los bits más significativos (MSBs) como resultado del desplazamiento a la derecha tienen un valor igual al MSB de x antes de la operación de desplazamiento.

x << y Desplazamiento aritmético a la izquierda de una representación entera en complemento a dos de x por y dígitos binarios. Esta función está definida solo para valores enteros no negativos de y. Los bits desplazados a los bits menos significativos (LSBs) como resultado del desplazamiento a la izquierda tienen un valor igual a 0.

Operadores de asignación

Los siguientes operadores aritméticos se definen de la siguiente manera:

= Operador de asignación

++ Incremento, es decir, x++ es equivalente a x = x + 1; cuando se usa en un índice de matriz, evalúa el valor de la variable antes de la operación de incremento.

-- Decremento, es decir, x-- es equivalente a x = x - 1; cuando se usa en un índice de matriz, evalúa el valor de la variable antes de la operación de disminución.

+= Incremento por la cantidad especificada, es decir, x += 3 es equivalente a x = x + 3, y x+ = (-3) es equivalente a x = x + (-3).

-= Decremento por la cantidad especificada, es decir, x -= 3 es equivalente a x = x - 3, y x -= (-3) es equivalente a x = x - (-3).

Notación de rango

La siguiente notación se utiliza para especificar un rango de valores:

x = y..z x toma valores enteros comenzando desde y hasta z, inclusive, siendo x, y, z números enteros y z mayor que y.

Funciones matemáticas

Se definen las siguientes funciones matemáticas:

$$\text{Abs}(x) = \begin{cases} x & ; \quad x \geq 0 \\ -x & ; \quad x < 0 \end{cases}$$

Asen(x) la función seno trigonométrica inversa, que opera sobre un argumento x que es en el rango de -1.0 a 1.0, inclusive, con un valor de salida en el rango de $-\pi \div 2$ a $\pi \div 2$, inclusive, en unidades de radianes

Atan(x) la función trigonométrica inversa tangente, que opera sobre un argumento x, con un valor de salida en el rango de $-\pi \div 2$ a $\pi \div 2$, inclusive, en unidades de radianes

$$5 \quad \text{Atan2}(y, x) = \begin{cases} \text{Atan}\left(\frac{y}{x}\right) & ; \quad x > 0 \\ \text{Atan}\left(\frac{y}{x}\right) + \pi & ; \quad x < 0 \ \&\& \ y \geq 0 \\ \text{Atan}\left(\frac{y}{x}\right) - \pi & ; \quad x < 0 \ \&\& \ y < 0 \\ +\frac{\pi}{2} & ; \quad x = 0 \ \&\& \ y \geq 0 \\ -\frac{\pi}{2} & ; \quad \text{de otra forma} \end{cases}$$

Ceil(x) el número entero más pequeño mayor o igual que x.

Clip1_v(x) = Clip3(0, (1 << BitDepth_v) - 1, x)

10

Clip1_c(x) = Clip3(0, (1 << BitDepth_c) - 1, x)

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{de otra forma} \end{cases}$$

15

Cos(x) la función coseno trigonométrica que opera sobre un argumento x en unidades de radianes.

Floor(x) el entero más grande menor o igual que x.

$$\text{GetCurrMsb}(a, b, c, d) = \begin{cases} c + d & ; \quad b - a \geq d / 2 \\ c - d & ; \quad a - b > d / 2 \\ c & ; \quad \text{de otra forma} \end{cases}$$

20

Ln(x) el logaritmo natural de x (el logaritmo base-e, donde e es el logaritmo natural base constante 2.718 281 828 ...).

Log2(x) el logaritmo en base 2 de x.

25

Log10(x) el logaritmo en base 10 de x.

$$\text{Min}(x, y) = \begin{cases} x & ; \quad x \leq y \\ y & ; \quad x > y \end{cases}$$

30

$$\text{Max}(x, y) = \begin{cases} x & ; \quad x \geq y \\ y & ; \quad x < y \end{cases}$$

Round(x) = Sign(x) * Floor(Abs(x) + 0.5)

$$\text{Sign}(x) = \begin{cases} 1 & ; \quad x > 0 \\ 0 & ; \quad x == 0 \\ -1 & ; \quad x < 0 \end{cases}$$

35

Sen(x) la función seno trigonométrica que opera sobre un argumento x en unidades de radianes

Sqrt(x) = $\sqrt{x}$

40

Swap(x, y) = (y, x)

Tan(x) la función tangente trigonométrica que opera sobre un argumento x en unidades de radianes

Orden de precedencia de la operación

45

Cuando un orden de precedencia en una expresión no se indica explícitamente mediante el uso de paréntesis, se aplican las siguientes reglas:

- Las operaciones de mayor precedencia se evalúan antes que cualquier operación de menor precedencia.

50

- Las operaciones de la misma precedencia se evalúan secuencialmente de izquierda a derecha.

La tabla 14 a continuación especifica la precedencia de las operaciones de mayor a menor; una posición más alta en la tabla 14 indica una precedencia más alta.

- 5 Para aquellos operadores que también se utilizan en el lenguaje de programación C, el orden de precedencia utilizado en esta descripción es el mismo que se utiliza en el lenguaje de programación C.

Tabla 14: Precedencia de operaciones desde la más alta (en la parte superior de la tabla) a la más baja (en la parte inferior de la tabla)

10

operaciones (con operandos x, y y z)
"x++", "x--"
"!x", "-x" (como operador de prefijo unario)
x^y
"x * y", "x / y", "x ÷ y", " $\frac{x}{y}$ ", "x % y"
"x + y", "x - y" (como operador de dos argumentos), " $\sum_{i=x}^y f(i)$ "
"x << y", "x >> y"
"x < y", "x <= y", "x > y", "x >= y"
"x == y", "x != y"
"x & y"
"x y"
"x && y"
"x y"
"x ? y : z"
"x..y"
"x = y", "x += y", "x -= y"

Descripción de texto de operaciones lógicas

En el texto, un enunciado de operaciones lógicas como se describiría matemáticamente de la siguiente forma:

15

si (condición 0)
 declaración 0
 sino si (condición 1)
 declaración 1

20

...
 sino / * comentario informativo sobre la condición restante * /
 declaración n

puede describirse de la siguiente manera:

25

... como sigue / ... se aplica lo siguiente:

- Si condición 0, declaración 0

30

- De lo contrario, si condición 1, declaración 1

— ...

- De lo contrario (comentario informativo sobre la condición restante), declaración n

35

Cada declaración "Si ... De lo contrario, si ... De lo contrario, ..." en el texto se introduce con "... como sigue" o "... se aplica lo siguiente" inmediatamente seguido de "Si ... ". La última condición del "Si ... De lo contrario, si ... De lo contrario, ..." es siempre un "De lo contrario, ...". Declaraciones intercalada "Si ... De lo contrario, si ... De lo contrario, ..." se pueden identificar haciendo coincidir "... como sigue" o "... se aplica lo siguiente" con la terminación "De lo contrario, ...".

40

En el texto, un enunciado de operaciones lógicas como se describiría matemáticamente de la siguiente forma:

si(condición 0a && condición 0b)
 declaración 0

45

sino si (condición 1a | condición 1b)
declaración 1

...
sino

5 declaración n

puede describirse de la siguiente manera:

... como sigue / ... se aplica lo siguiente:

10 - Si se cumplen todas las condiciones siguientes, enunciado 0:

- condición 0a

15 - condición 0b

- De lo contrario, si una o más de las siguientes condiciones son verdaderas, declaración 1:

- condición 1a

20 - condición 1b

— ...

25 - De lo contrario, declaración n

En el texto, un enunciado de operaciones lógicas como se describiría matemáticamente de la siguiente forma:

30 si (condición 0)
 declaración 0
 si (condición 1)
 declaración 1

puede describirse de la siguiente manera:

35 Cuando la condición 0, declaración 0

 Cuando la condición 1, declaración 1

40 Aunque las realizaciones de la invención se han descrito principalmente en base a la codificación de video, debe tenerse en cuenta que las realizaciones del sistema de codificación 10, el codificador 20 y el decodificador 30 (y en consecuencia el sistema 10) y las otras realizaciones descritas en este documento también pueden configurarse para procesamiento o codificación de imágenes, es decir, el procesamiento o codificación de una imagen individual independientemente de cualquier imagen anterior o consecutiva como en la codificación de video. En general, solo
45 las unidades de interpredicción 244 (codificador) y 344 (decodificador) pueden no estar disponibles en caso de que la codificación de procesamiento de imágenes se limite a una sola imagen 17. Todas las demás funcionalidades (también denominadas herramientas o tecnologías) del codificador de video 20 y del decodificador de video 30 pueden usarse igualmente para el procesamiento de imágenes fijas, por ejemplo, cálculo residual 204/304, transformada 206, cuantificación 208, cuantificación inversa 210/310, transformación (inversa) 212/312, partición 262/362, intrapredicción 254/354 y/o filtrado de bucle 220, 320 y codificación de entropía 270 y decodificación de entropía 304.

Realizaciones, por ejemplo, del codificador 20 y el decodificador 30, y las funciones descritas en el presente documento, por ejemplo, con referencia al codificador 20 y al decodificador 30, pueden implementarse en hardware,
55 software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse en un medio legible por ordenador o transmitirse a través de medios de comunicación como una o más instrucciones o código y ejecutarse mediante una unidad de procesamiento basada en hardware. El medio legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible como los medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que
60 facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, según un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medios de almacenamiento legibles por ordenador tangibles que no son transitorios o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que puedan acceder una o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o
65 estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa de ordenador puede incluir un medio legible por ordenador.

A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético, u otros dispositivos de almacenamiento magnético, memoria flash, o cualquier otro medio que pueda utilizarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda acceder un ordenador. Además, cualquier conexión se denomina propiamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota utilizando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de suscriptor digital (DSL) o tecnologías inalámbricas como los infrarrojos, la radio y las microondas, el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas como los infrarrojos, la radio y las microondas se incluyen en la definición de medio. Debe entenderse, sin embargo, que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que se dirigen a medios de almacenamiento tangibles no transitorios. Disquete y disco, tal y como se utilizan aquí, incluyen el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco floppy y el disco Blu-ray, donde los disquetes suelen reproducir los datos magnéticamente, mientras que los discos reproducen los datos ópticamente con láser. Las combinaciones de lo anterior también deben incluirse en el ámbito de los medios legibles por ordenador.

Las instrucciones pueden ser ejecutada por uno o más procesadores, como uno o más procesadores de señales digitales (DSPs), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASICs), arreglos de lógica programable en campo (FPGAs) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", tal y como se utiliza en el presente documento, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados, configurados para la codificación y decodificación, o incorporados en un códec combinado. Además, todas las tecnologías pueden implementarse en uno o más circuitos o elementos lógicos.

Las técnicas de esta descripción pueden implementarse en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de ICs (por ejemplo, un conjunto de chips). En esta descripción se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de los dispositivos configurados para llevar a cabo las técnicas divulgadas, pero no requieren necesariamente la realización de diferentes unidades de hardware. Más bien, como se describió anteriormente, varias unidades pueden combinarse en una unidad de hardware de códec o proporcionarse mediante una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describió anteriormente, junto con software y/o firmware adecuados.

REIVINDICACIONES

1. Un método de intrapredicción para un bloque de imagen implementado por un dispositivo de decodificación, caracterizado porque comprende:

obtener (1201, 1213) un valor de una primera información de indicación de un bloque actual a partir de un flujo de bits cuando está presente la primera información de indicación, o se infiere que el valor de la primera información de indicación es igual a 1 cuando la primera información de indicación no está presente en el flujo de bits, el valor de la primera información de indicación que indica si un modo de intrapredicción del bloque actual está comprendido en un conjunto de modos más probables, donde el conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y modo planar;

obtener un valor de una línea de índice de referencia del bloque actual a partir del flujo de bits cuando está presente el valor de la línea de índice de referencia, o se infiere que el valor de la línea de índice de referencia es igual a 0 cuando la línea de índice de referencia no está presente en el flujo de bits;

cuando el valor de la primera información de indicación indica que el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y cuando el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual, obtener (1205, 1214) un valor de una segunda información de indicación del bloque actual a partir del flujo de bits cuando está presente la segunda información de indicación, o se infiere que el valor de la segunda información de indicación es igual a 1 cuando la segunda información de indicación no está presente en el flujo de bits, el valor de la segunda información de indicación del bloque actual indica si el modo de intrapredicción del bloque actual es el modo planar o no, caracterizado porque la primera información de indicación se indica mediante una primera bandera, `intra_luma_mpm_flag`, y la segunda información de indicación se indica mediante una segunda bandera, `intra_luma_not_planar_flag`, caracterizado porque cuando el valor de `intra_luma_mpm_flag` es igual a 1, el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y cuando el valor de la línea de índice de referencia del bloque actual es igual a 0, se usa la línea de referencia vecina más cercana al bloque actual,

cuando el valor de la segunda información de indicación del bloque actual que indica el modo de intrapredicción del bloque actual no es el modo planar, se analiza `intra_luma_mpm_idx` correspondiente a un índice de los 5 modos de intrapredicción candidatos, que predicen el bloque actual de acuerdo con el índice y las entradas de una lista MPM.

2. El método de la reivindicación 1, caracterizado porque el valor de la segunda información de indicación está codificado por contexto de codificación aritmética binaria adaptable al contexto (CABAC).

3. El método de cualquiera de las reivindicaciones 1 a 2, caracterizado porque el método comprende además:

analizar la segunda bandera usando el contexto en base a si el bloque actual es un intra bloque con el modo de codificación de intra subpartición (ISP) habilitado o no.

4. El método de cualquiera de las reivindicaciones 1 a 3, caracterizado porque cuando el valor de la segunda información de indicación del bloque actual indica que el modo de intrapredicción del bloque actual no es el modo planar, y cuando una o más de las siguientes condiciones son verdaderas, un modo de intrapredicción candidato `candIntraPredModeX` se establece igual a `INTRA_PLANAR`:

- La variable `availableX` es igual a FALSO;
- `CuPredMode[xNbX][yNbX]` no es igual a `MODE_INTRA`; o
- X es igual a B y $yCb - 1$ es menor que $((yCb \gg CtbLog2SizeY) \ll CtbLog2SizeY)$.

donde X se reemplaza ya sea por el bloque vecino A o B , una ubicación de luma (xCb , yCb) que especifica la muestra superior izquierda del bloque de codificación de luma actual en relación con la muestra de luma superior-izquierda de la imagen actual, la ubicación ($xNbA$, $yNbA$) del vecino A se establece igual a ($xCb - 1$, $yCb + cbHeight - 1$), la ubicación ($xNbB$, $yNbB$) del vecino B se establece igual a ($xCb + cbWidth - 1$, $yCb - 1$), una variable `cbWidth` que especifica el ancho del bloque de codificación actual en muestras de luma, una variable `cbHeight` que especifica la altura del bloque de codificación actual en muestras de luma, `CuPredMode` representa un modo de predicción actual, `SizeY` representa el tamaño de los componentes Y de un bloque de árbol de codificación.

5. El método de cualquiera de las reivindicaciones 1 a 4, caracterizado porque cuando un modo de intrapredicción de un bloque vecino izquierdo y un modo de intrapredicción de un bloque vecino superior son ambos modos no angulares, la lista MPM comprende:

`candModeList[0] = INTRA_DC`

candModeList[1] = INTRA_ANGULAR50

candModeList[2] = INTRA_ANGULAR18

5 candModeList[3] = INTRA_ANGULAR46

candModeList[4] = INTRA_ANGULAR54.

10 6. El método de cualquiera de las reivindicaciones 1 a 5, caracterizado porque el paso de obtener el valor de la segunda información de indicación comprende:

analizar un flujo de bits para obtener el valor de la segunda información de indicación;

15 o cuando se habilita la codificación de línea de referencia múltiple (MRL), el valor de la segunda información de indicación se obtiene mediante un valor predeterminado.

7. Un método de intrapredicción para un bloque de imagen implementado por un dispositivo de codificación, caracterizado porque comprende:

20 determinar (1301) si un modo de intrapredicción de un bloque actual está comprendido en un conjunto de modos más probables, en donde el conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y modo planar;

determinar (1302) si la línea de referencia vecina más cercana al bloque actual se usa para la intrapredicción;

25 codificar (1303) un flujo de bits, en donde el flujo de bits incluye información para indicar si el modo de intrapredicción del bloque actual es el modo planar cuando el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y la línea de referencia vecina más cercana al bloque actual se utiliza para la intrapredicción, caracterizado porque el flujo de bits incluye una primera bandera intra_luma_mpm_flag que indica si el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y una segunda bandera intra_luma_not_planar_flag que indica si el modo de intrapredicción del bloque actual es el modo planar; caracterizado porque cuando el valor de intra_luma_mpm_flag es igual a 1, el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y cuando un valor de la línea de índice de referencia del bloque actual es igual a 0, se usa la línea de referencia vecina más cercana al bloque actual;

35 cuando el valor de la segunda bandera que indica el modo de intrapredicción del bloque actual no es el modo planar, el flujo de bits incluye intra_luma_mpm_idx correspondiente a un índice de los 5 modos de intrapredicción candidatos.

40 8. El método de la reivindicación 7, caracterizado porque la información para indicar si el modo de intrapredicción del bloque actual está codificado por contexto de codificación aritmética binaria adaptable al contexto (CABAC), y caracterizado porque el contexto incluye si el bloque actual es un intra bloque con el modo de codificación de intra subpartición (ISP) habilitado.

45 9. Un dispositivo de decodificación para intrapredicción de un bloque de imagen, el dispositivo caracterizado porque comprende:

una primera unidad de obtención (1401), configurada para obtener un valor de una línea de índice de referencia de un bloque actual a partir de un flujo de bits cuando está presente el valor de la línea de índice de referencia, o se infiere que el valor de la línea de índice de referencia es igual a 0 cuando la línea de índice de referencia no está presente en el flujo de bits, y configurada además para obtener un valor de una primera información de indicación del bloque actual a partir de flujo de bits cuando está presente la primera información de indicación, o se infiere que el valor de la primera información de indicación es igual a 1 cuando la primera información de indicación no está presente en el flujo de bits, el valor de la primera información de indicación indica si un modo de intrapredicción del bloque actual está comprendido en un conjunto de modos más probables, en donde el conjunto de modos más probables comprende 5 modos de intrapredicción candidatos y modo planar;

una segunda unidad de obtención (1402), configurada para obtener un valor de una segunda información de indicación del bloque actual a partir de un flujo de bits cuando está presente la segunda información de indicación, o se infiere que el valor de la segunda información de indicación es igual a 1 cuando la segunda información de indicación no está presente en el flujo de bits, cuando el valor de la primera información de indicación indica que el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y cuando el valor de la línea de índice de referencia indica que se usa la línea de referencia vecina más cercana al bloque actual, en donde el valor de la segunda información de indicación del bloque actual indica si el modo de intrapredicción del bloque actual es el modo planar, caracterizado porque la primera información de indicación se indica mediante una primera bandera intra_luma_mpm_flag, y la segunda información de indicación se indica mediante una segunda bandera intra_luma_not_planar_flag, caracterizado porque cuando el valor de intra_luma_mpm_flag es igual a 1, el

modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y cuando el valor de la línea de índice de referencia del bloque actual es igual a 0, se utiliza la línea de referencia vecina más cercana al bloque actual;

5 cuando el valor de la segunda información de indicación del bloque actual que indica el modo de intrapredicción del bloque actual no es el modo planar, se analiza `intra_luma_mpm_idx` correspondiente a un índice de los 5 modos de intrapredicción candidatos, en donde se predice el bloque actual de acuerdo con el índice y las entradas de una lista MPM.

10 10. Un dispositivo de codificación para intrapredicción de un bloque de imagen, caracterizado porque comprende:

una unidad de determinación (1501), configurada para determinar si un modo de intrapredicción de un bloque actual está comprendido en un conjunto de modos más probables, y para determinar si la línea de referencia vecina más cercana al bloque actual se usa para intrapredicción, donde el conjunto de los modos más probables comprende 5

15 modos de intrapredicción candidatos y el modo planar;

una unidad de codificación (1502), configurada para codificar un flujo de bits, en donde el flujo de bits incluye información para indicar si el modo de intrapredicción del bloque actual es el modo planar cuando el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y la línea de referencia

20 vecina más cercana al bloque actual se utiliza para la intrapredicción;

caracterizado porque el flujo de bits incluye una primera bandera `intra_luma_mpm_flag` que indica si el modo de intrapredicción del bloque actual está comprendido en el conjunto de modos más probables, y una segunda bandera `intra_luma_not_planar_flag` que indica si el modo de intrapredicción del bloque actual es el modo planar;

25 cuando el valor de la segunda bandera que indica el modo de intrapredicción del bloque actual no es el modo planar, el flujo de bits incluye `intra_luma_mpm_idx` correspondiente a un índice de los 5 modos de intrapredicción candidatos;

30 en donde se predice el bloque actual de acuerdo con el índice y las entradas de una lista MPM.

11. Un producto de programa de ordenador que comprende un código de programa que, cuando es ejecutado por un ordenador, hace que el ordenador realice el método de acuerdo con cualquiera de las reivindicaciones 1 a 8.

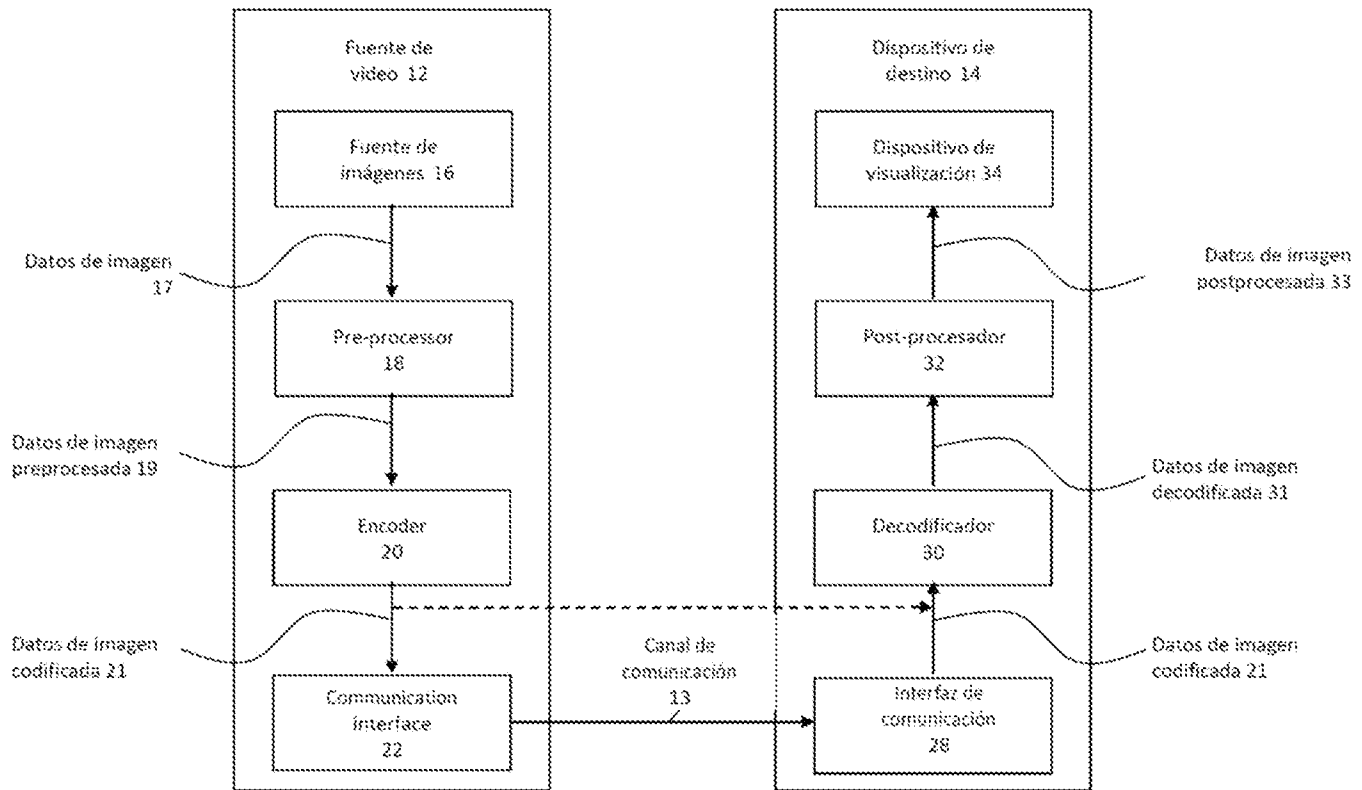


FIG. 1A

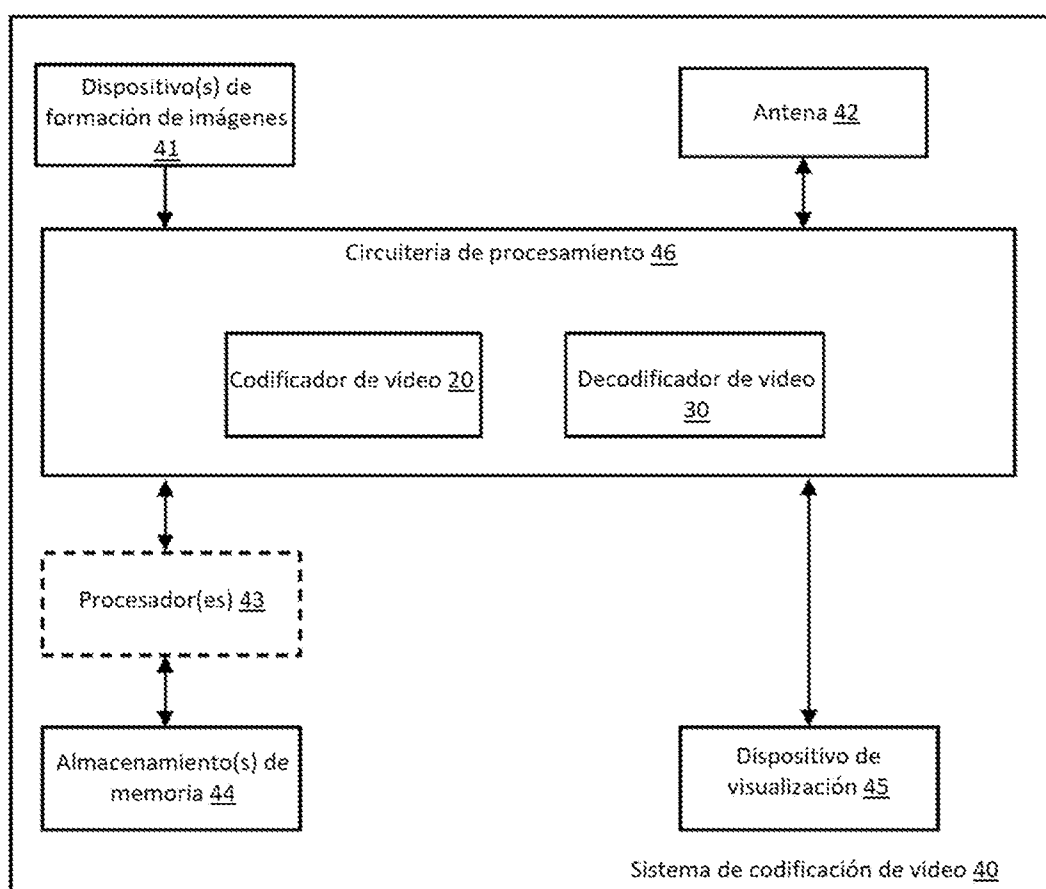


FIG. 1B

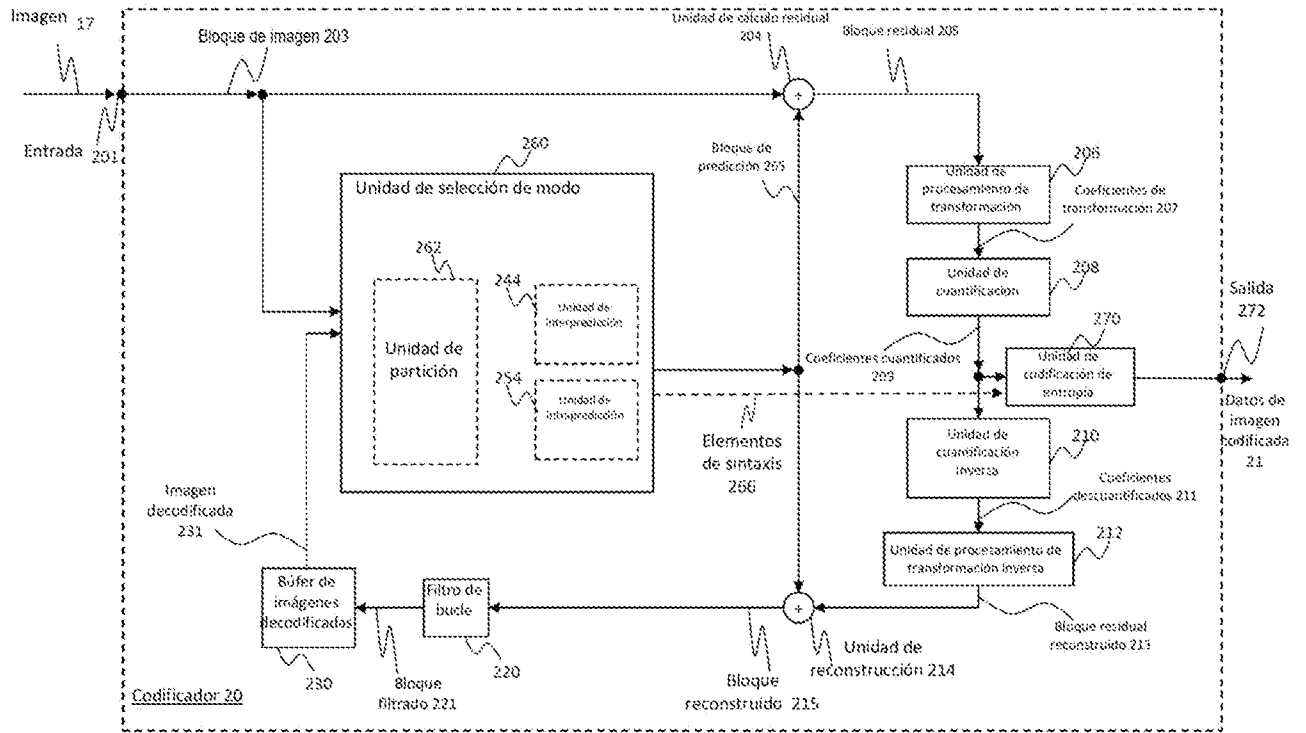
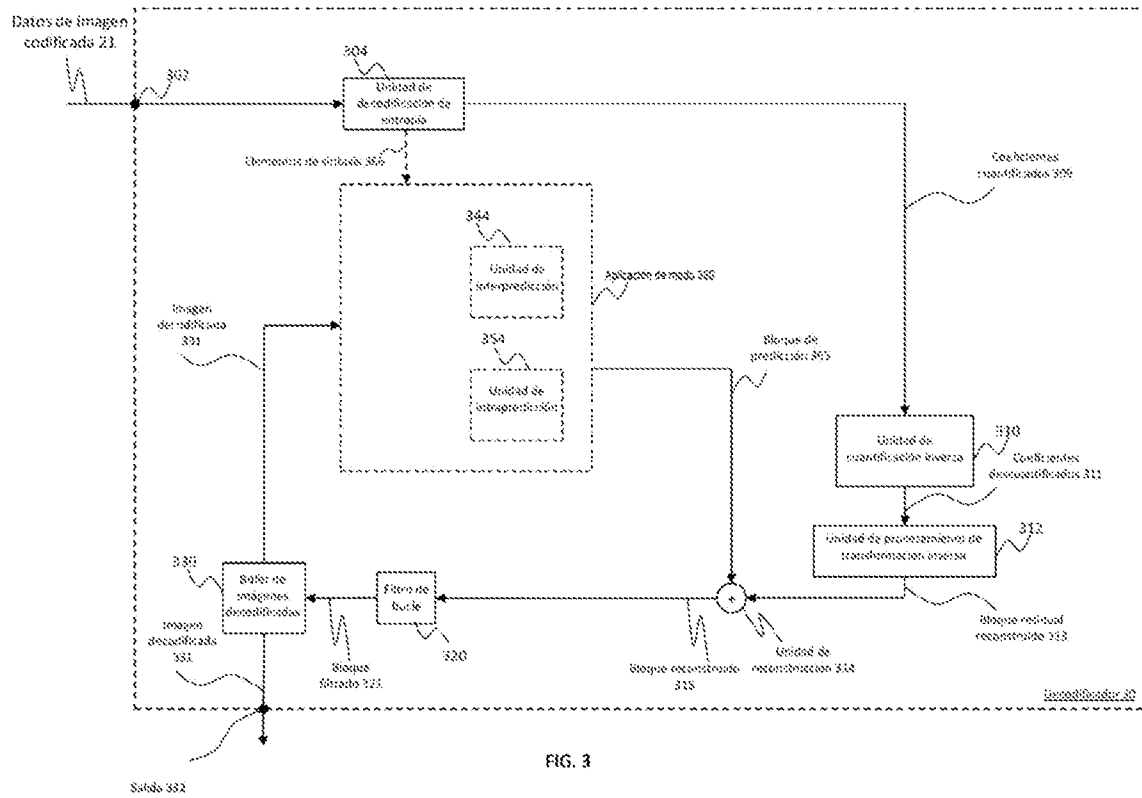


FIG. 2



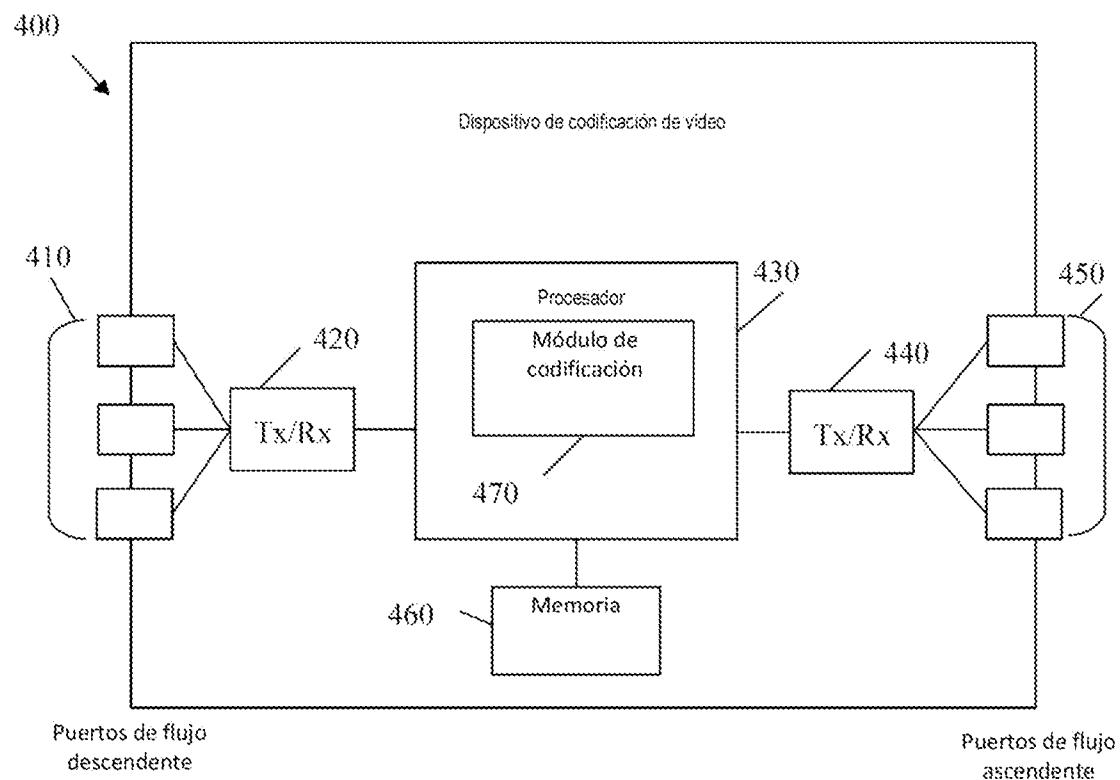


FIG. 4

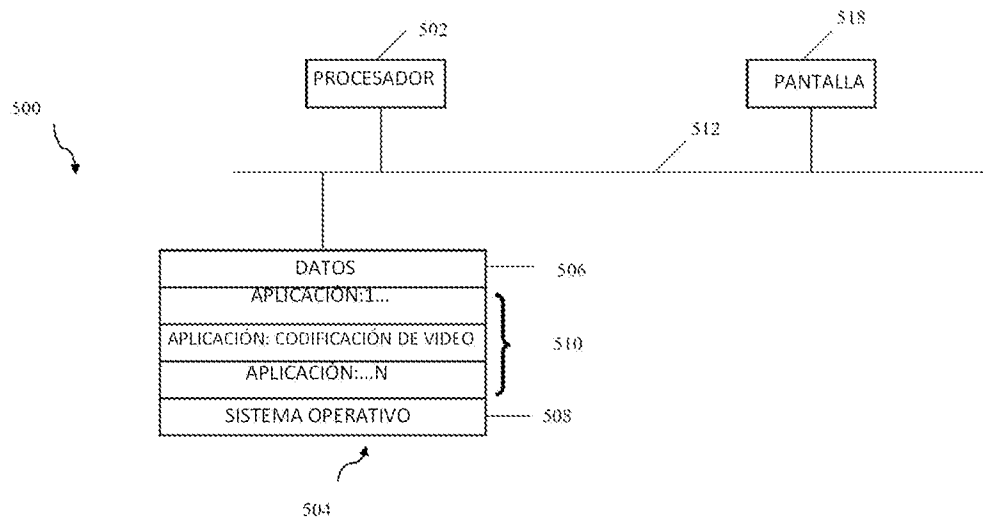


FIG. 5

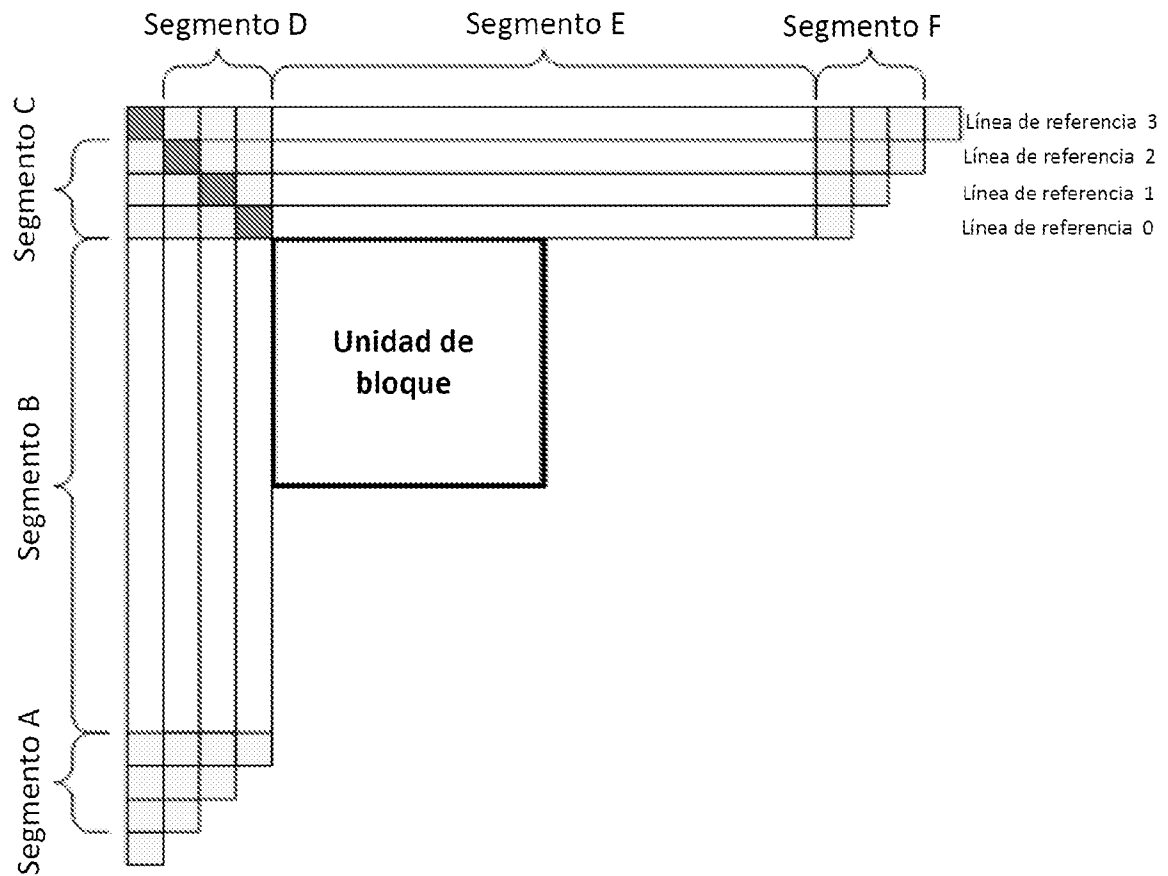


FIG. 6, Herramienta de codificación de línea de referencia, indicación de líneas de referencia vecinas. Cuando el índice de línea de referencia es igual a 0, la línea de referencia vecina más cercana al bloque actual es utilizada.

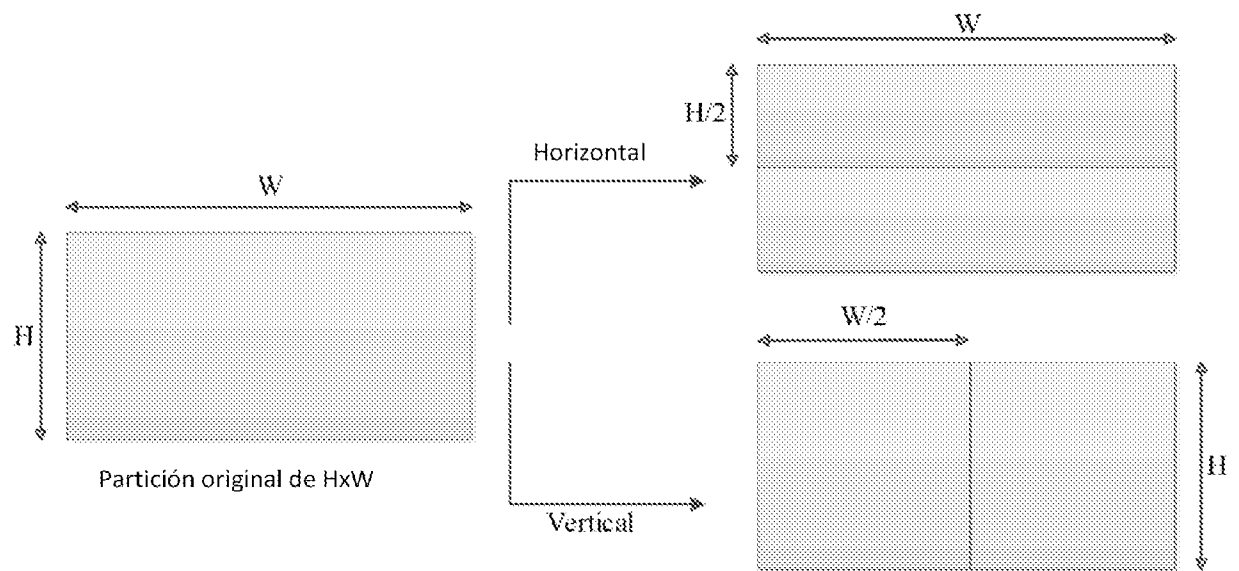


FIG. 7: Ejemplo de división de bloques de 4 x 8 y 8 x 4

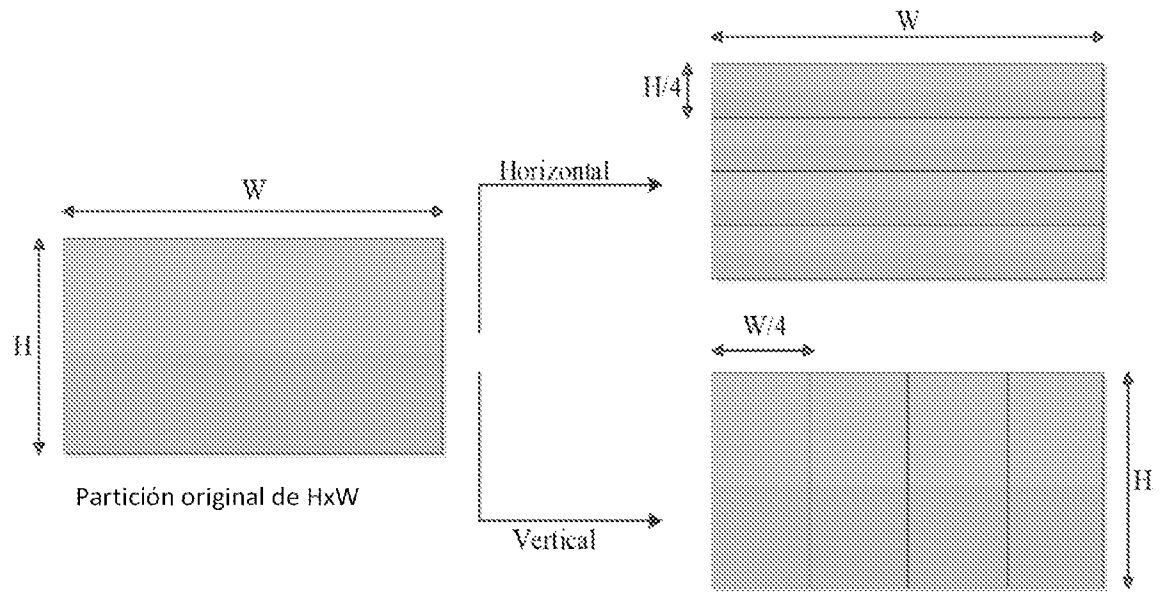


FIG. 8: Ejemplo de división de todos los bloques excepto 4x8, 8x4 y 4x4

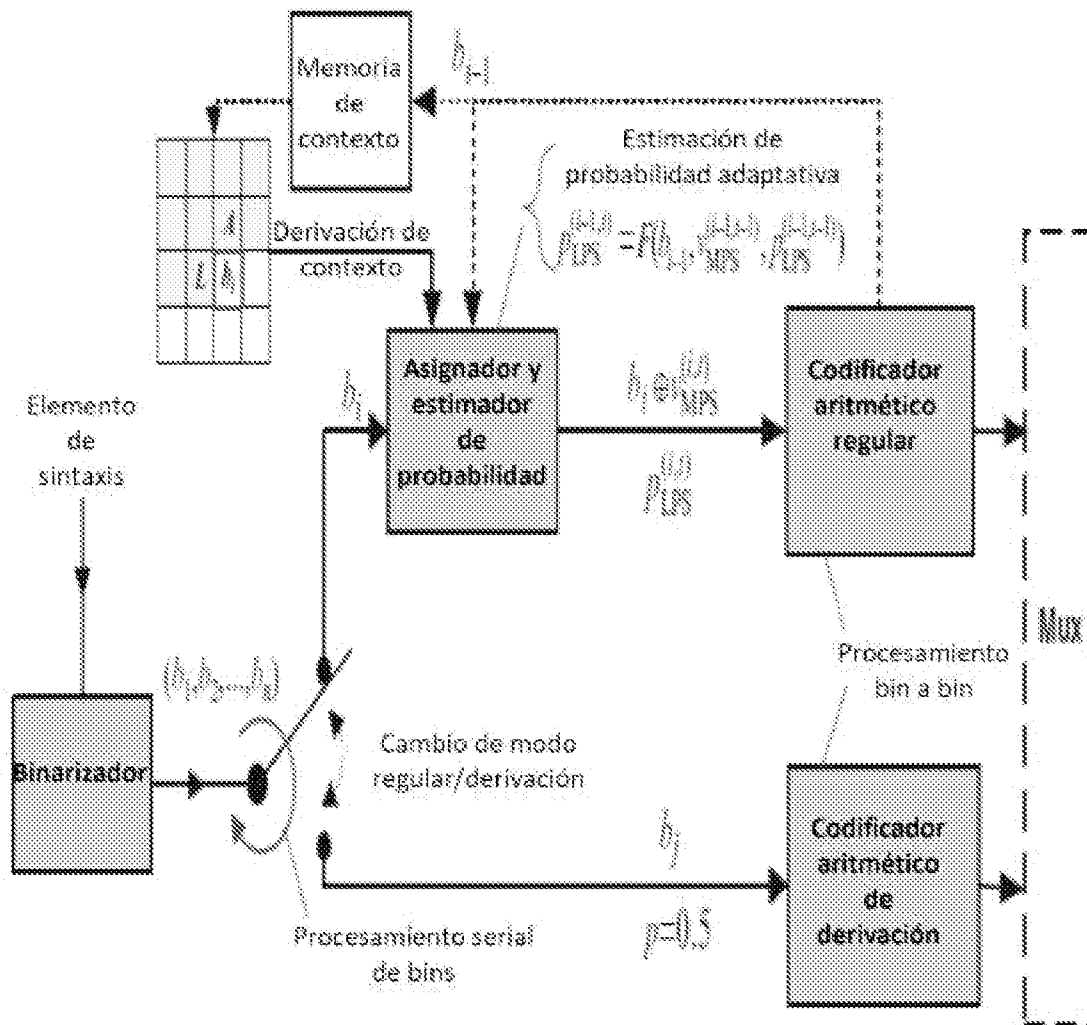


FIG. 9

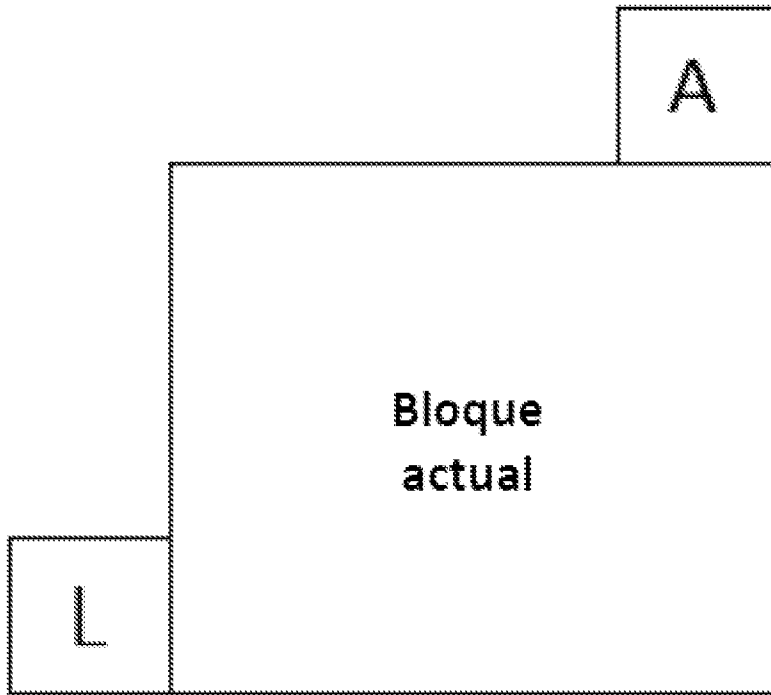


FIG. 10

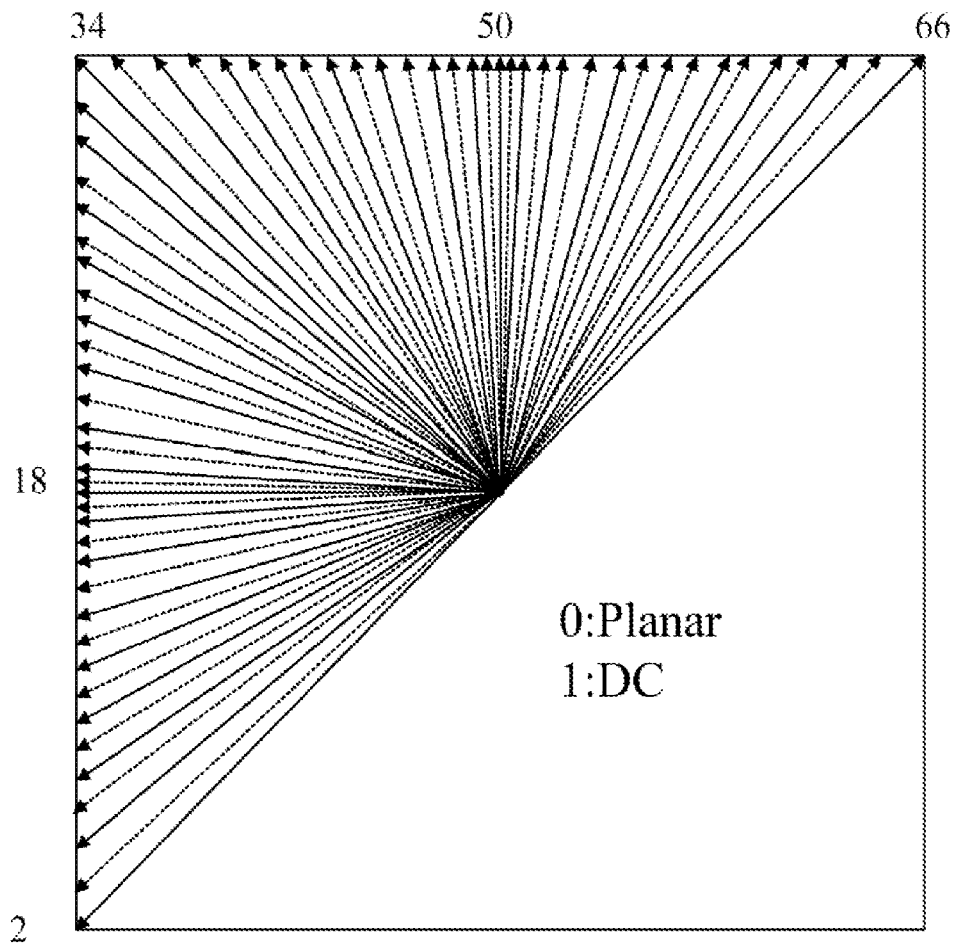


FIG. 11

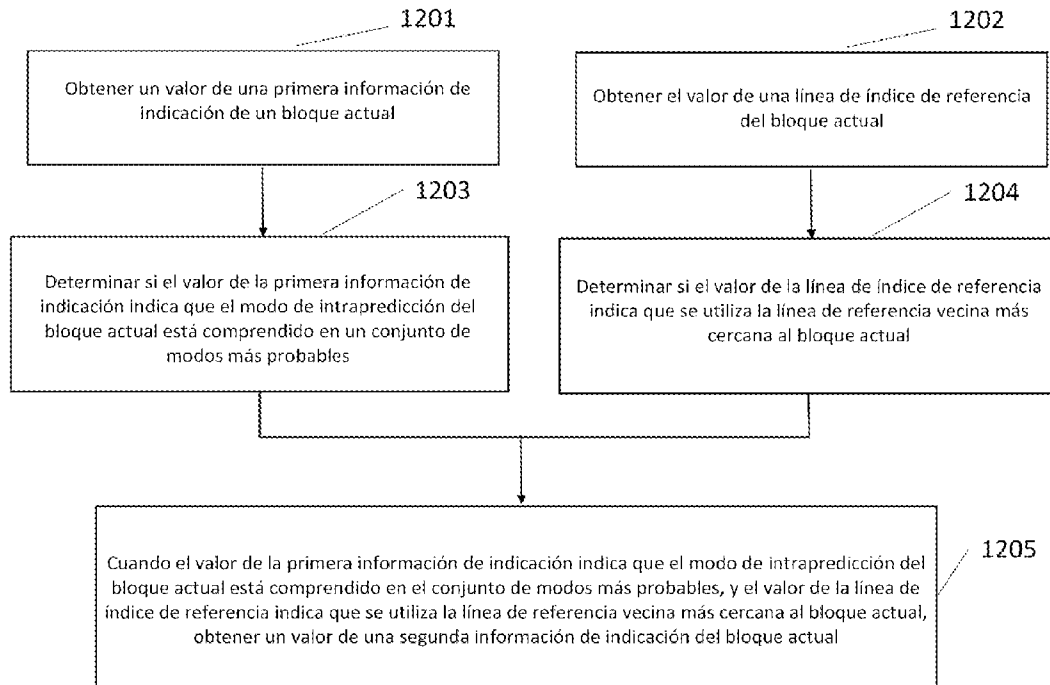


FIG. 12A

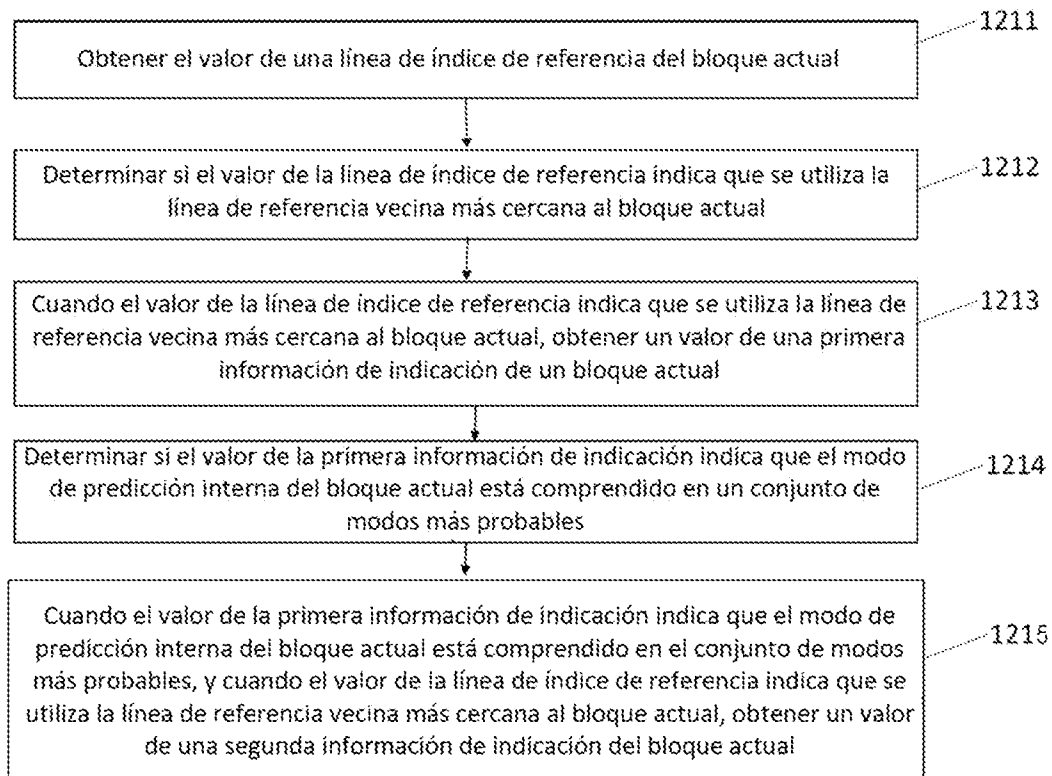


FIG. 12B

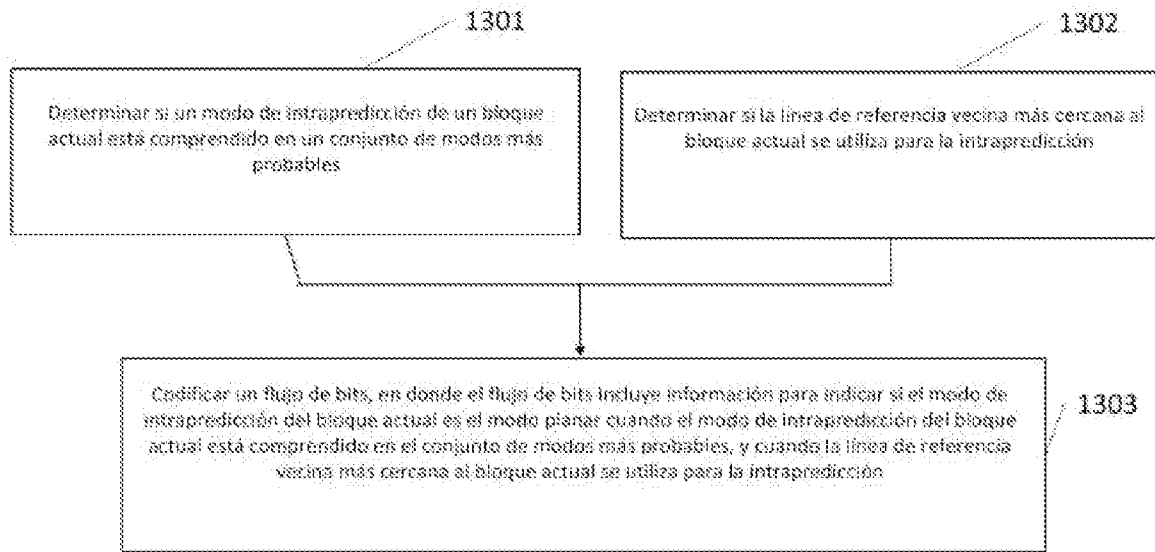


FIG. 13

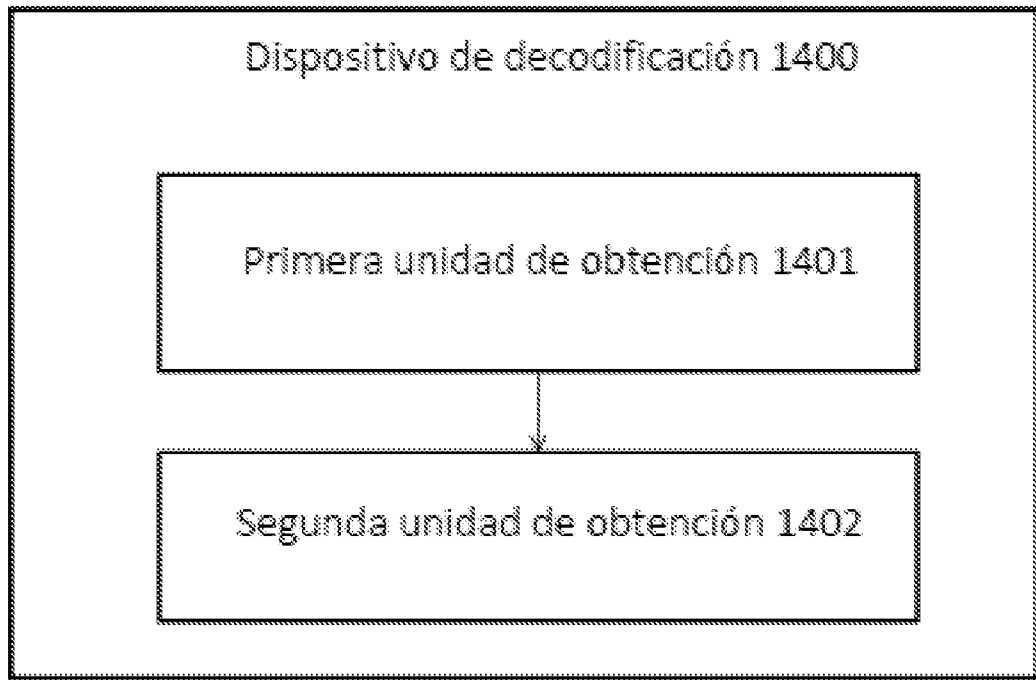


FIG. 14

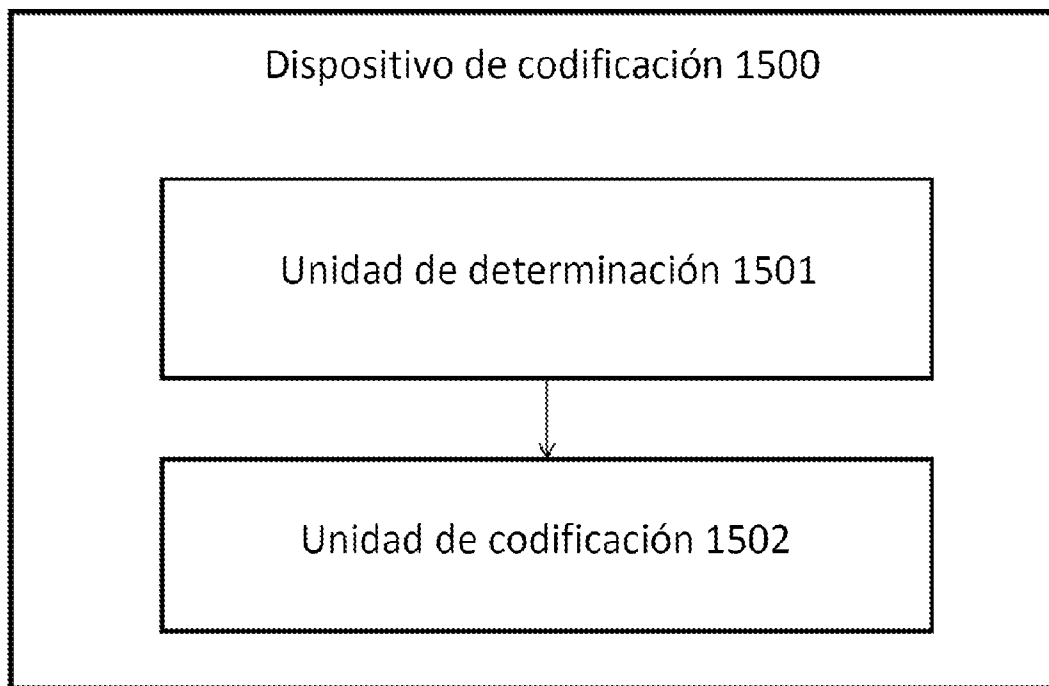


FIG. 15

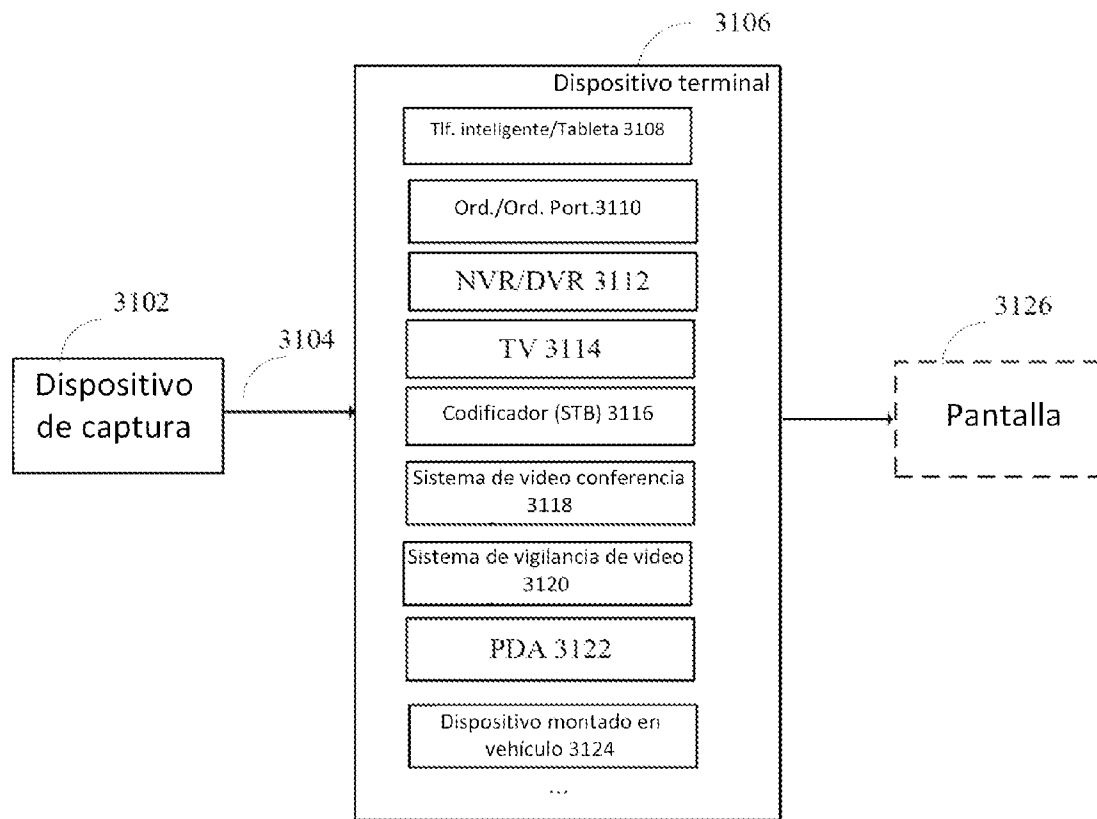


FIG. 16

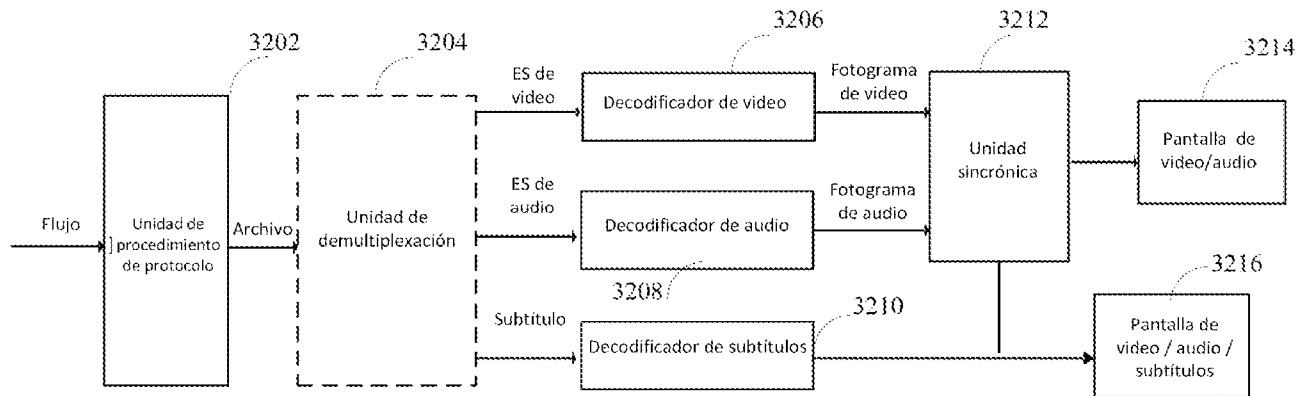


FIG. 17