

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 987 742**

51 Int. Cl.:

H04N 19/105 (2014.01)

H04N 19/157 (2014.01)

H04N 19/176 (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/RU2020/050055**

87 Fecha y número de publicación internacional: **24.09.2020** **WO20190179**

96 Fecha de presentación y número de la solicitud europea: **23.03.2020** **E 20773813 (9)**

97 Fecha y número de publicación de la concesión europea: **31.07.2024** **EP 3932056**

54 Título: **Método y aparato para intrapredicción**

30 Prioridad:

21.03.2019 US 201962822047 P

11.04.2019 US 201962832878 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
18.11.2024

73 Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.0%)

Huawei Administration Building, Bantian,

Longgang District,

Shenzhen, Guangdong 518129, CN

72 Inventor/es:

RUFITSKIY, VASILY ALEXEEVICH;

WANG, BIAO;

ESENLIK, SEMIH;

CHEN, JIANLE y

FILIPPOV, ALEXEY KONSTANTINOVICH

74 Agente/Representante:

PONS ARIÑO, Ángel

ES 2 987 742 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

Método y aparato para intrapredicción

5 Campo técnico

Las realizaciones de la presente solicitud (descripción) generalmente se relacionan con el campo del procesamiento de imágenes y más particularmente con la intrapredicción, especialmente con el método y aparato para el modo planar o DC usado en la intrapredicción, tal como intrasubpartición o intrapredicción de líneas de referencia múltiple.

Antecedentes

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

La cantidad de datos de vídeo necesarios para representar incluso un vídeo relativamente corto puede ser sustancial, lo que puede generar dificultades cuando los datos se van a transmitir o comunicar de otro modo a través de una red de comunicaciones con capacidad de ancho de banda limitada. Por lo tanto, los datos de vídeo generalmente se comprimen antes de comunicarse a través de las redes de telecomunicaciones modernas. El tamaño de un vídeo también podría ser un problema cuando el vídeo se almacena en un dispositivo de almacenamiento debido a que los recursos de memoria pueden ser limitados. Los dispositivos de compresión de vídeo a menudo usan software y/o hardware en el origen para codificar los datos de vídeo antes de la transmisión o el almacenamiento, lo que reduce la cantidad de datos necesarios para representar imágenes de vídeo digital. Luego, los datos comprimidos se reciben en el destino por un dispositivo de descompresión de vídeo que decodifica los datos de vídeo. Con recursos de red limitados y demandas cada vez mayores de mayor calidad del vídeo, son deseables técnicas mejoradas de compresión y descompresión que mejoren la relación de compresión con poco o ningún sacrificio en la calidad de imagen.

El documento WO 2017/222325A1 describe un método de procesamiento de señales de vídeo.

El documento de B. Bross et al. "Versatile vídeo Coding (Draft 2)", 11. JVET MEETING; 20107103 - 20180718; Ljubljana; no. JVET-K1001-v6 presenta un borrador de una especificación de codificación de vídeo.

El documento de ALBRECHT M ET AL: "Description of SDR, HDR and 360° vídeo coding technology proposal by Fraunhofer HHI", 10. JVET MEETING; 20180410 - 20180420; SAN DIEGO; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), no. JVET-J0014 12 de abril de 2018, describe aplicar MRL al intramodo DC.

Además, el documento de BROSS (FRAUNHOFER) B ET AL: "CE3: Multiple reference line intra prediction (Test 1.1.1, 1.1.2, 1.1.3 and 1.1.4)", 12. JVET MEETING; 20181003 - 20181012; MACAO; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), no. JVET-L0283 de octubre de 2018 enseña el uso de la predicción de líneas de muestreo de referencia superior e izquierda para bloques cuadrados y la predicción de líneas de muestreo de referencia superior o izquierda (XOR) única para bloques no cuadrados para Intrapredicción DC.

Compendio

Las realizaciones de la presente solicitud proporcionan aparatos y métodos para codificar y decodificar según las reivindicaciones independientes.

El objeto de las reivindicaciones independientes permite lograr los objetivos anteriores y otros. Otras formas de implementación se desprenden de las reivindicaciones dependientes, de la descripción y de las figuras.

Esta descripción describe sistemas y técnicas para la codificación y decodificación de vídeo, y específicamente un método y aparato para la intrapredicción usando un modo planar o DC.

En general, en un aspecto innovador, un método incluye

- selección de un filtro de muestras de referencia basado en la combinación de herramientas de intrapredicción, que comprende: ISP o predicción de línea de referencia múltiple

- intrapredicción de un bloque

- aplicar condicionalmente PDPC al resultado de la intrapredicción.

La presente descripción proporciona un método de intrapredicción del modo de intrapredicción DC para un bloque, que comprende: obtener un valor de un índice de línea de referencia de intrapredicción, $refIdx$, de un bloque de transformación; y

obtener muestras de predicción del bloque de transformación basándose en muestras de referencia $p[x][y]$ de una línea de referencia, en donde la línea de referencia está indexada por el valor del índice de línea de referencia de intrapredicción $refIdx$, siendo el valor de $refIdx$ distinto de cero;

en donde las muestras de predicción del bloque de transformación $predSamples[x][y]$ se obtienen mediante:

$predSamples[x][y] = dcVal$, con $x = 0..nTbW-1$, $y = 0..nTbH-1$

en donde cuando $nTbW$ es igual a $nTbH$, la variable $dcVal$ se obtiene mediante:

$$dcVal = (\sum_{x'=0}^{nTbW-1} p[x'][-1-refIdx] + \sum_{y'=0}^{nTbH-1} p[-1-refIdx][y'] + nTbW) \gg (\text{Log2}(nTbW) + 1);$$

en donde cuando el ancho del bloque de transformación es mayor que la altura del bloque de transformación, solo se usan muestras de referencia de una línea de referencia superior para obtener las muestras de predicción, y la variable $dcVal$ se obtiene mediante:

$$dcVal = (\sum_{x'=0}^{nTbW-1} p[x'][-1-refIdx] + (nTbW \gg 1)) \gg \text{Log2}(nTbW);$$

en donde cuando la altura del bloque de transformación es mayor que la anchura del bloque de transformación, solo se usan muestras de referencia de una línea de referencia izquierda para obtener las muestras de predicción, y la variable $dcVal$ se obtiene mediante:

$$dcVal = (\sum_{y'=0}^{nTbH-1} p[-1-refIdx][y'] + (nTbH \gg 1)) \gg \text{Log2}(nTbH);$$

$p[x'][-1-refIdx]$ que especifica las muestras de referencia desde una línea de referencia superior,

$p[-1-refIdx][y']$ que especifica las muestras de referencia desde una línea de referencia izquierda,

una variable $nTbW$ que especifica un ancho del bloque de transformación,

una variable $nTbH$ que especifica una altura del bloque de transformación;

y

en donde las muestras de referencia se determinan especificando un índice horizontal o vertical, cada uno de los índices es un valor de número entero que pertenece a un intervalo de valores de números enteros, y en donde los índices especifican una coordenada horizontal (x') o una coordenada vertical (y') dentro la línea de referencia, respectivamente; y

en donde la muestra de referencia más superior de la línea de referencia se desplaza verticalmente a la posición de la muestra predicha superior izquierda del bloque de transformación, y en donde el valor del desplazamiento es igual al valor del índice de línea de referencia de intrapredicción; y

la muestra de referencia más izquierda de la línea de referencia se desplaza horizontalmente a la posición de la muestra predicha superior izquierda del bloque de transformación, y en donde el valor del desplazamiento es igual al valor del índice de línea de referencia de intrapredicción.

La línea de referencia en la solicitud incluye no solo una fila de muestras de referencia, sino también una columna de muestras de referencia. Una fila de muestras de referencia es un ejemplo de una línea de referencia. Una columna de muestras de referencia es otro ejemplo de una línea de referencia.

La codificación de vídeo del estado de la técnica (por ejemplo, Codificación de Vídeo Versátil) comprende modos de intrapredicción direccional, DC y planar que se señalan usando el concepto de lista de modos más probables (MPM) también conocido en el borrador de especificación de VVC como modo candidato de intrapredicción $candModelist$. Cuando se usa MRL (el índice de línea de referencia no es cero), podría ocurrir que los modos de intrapredicción DC y Planar usen diferentes posiciones de las muestras de referencia dentro

de la línea de referencia seleccionada. La presente descripción introduce una armonización de MRL y de intrapredicción no direccional. La armonización consiste en alinear los conjuntos de muestras de referencia con la posición de la muestra superior izquierda del bloque de transformación.

5 Uno de los fundamentos detrás de esta invención es la unificación de la derivación de la lista MPM (modo más probable) que permite representar tanto el modo DC como el modo planar en la lista MPM. Esta unificación hace que el procesamiento de modos señalados en MPM sea más regular reduciendo el número total de ramas condicionales, lo que es beneficioso tanto para implementaciones de hardware como de software.

10 En el método descrito anteriormente, el valor del índice de línea de referencia de intrapredicción puede indicarse mediante `intra_luma_ref_idx`.

En el método descrito anteriormente, el valor del índice de línea de referencia de intrapredicción puede ser 0, 1 o 2.

15 En el método descrito anteriormente, las muestras de referencia pueden formar un intervalo contiguo o una distancia contigua dentro de la línea de referencia.

20 La presente descripción proporciona además un decodificador, que comprende: uno o más procesadores; y un soporte de almacenamiento no transitorio legible por ordenador acoplado a los procesadores y que almacena programación para su ejecución por los procesadores, en donde la programación, cuando es ejecutada por los procesadores, configura el decodificador para llevar a cabo el método descrito anteriormente.

25 La presente descripción proporciona además un codificador, que comprende: uno o más procesadores; y un soporte de almacenamiento no transitorio legible por ordenador acoplado a los procesadores y que almacena programación para su ejecución por los procesadores, en donde la programación, cuando es ejecutada por los procesadores, configura el codificador para llevar a cabo el método descrito anteriormente.

30 La presente descripción proporciona además un codificador que comprende un circuito de procesamiento para llevar a cabo el método descrito anteriormente.

La presente descripción proporciona además un decodificador que comprende un circuito de procesamiento para llevar a cabo el método descrito anteriormente.

35 La presente descripción proporciona además un producto de programa informático que comprende un código de programa para realizar el método descrito anteriormente.

40 La presente descripción proporciona además un soporte no transitorio legible por ordenador que transporta un código de programa que, cuando se ejecuta por un dispositivo informático, hace que el dispositivo informático realice el método descrito anteriormente. Los detalles de una o más realizaciones se exponen en los dibujos adjuntos y la descripción que sigue. Otras características, objetos y ventajas serán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

Breve descripción de los dibujos

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

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

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

55 La FIG. 1C es un diagrama de bloques que muestra un ejemplo de un dispositivo móvil configurado para implementar realizaciones de la invención;

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

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

65 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;

5 La FIG. 6 es una ilustración de dos subparticiones divididas en la herramienta ISP;

La FIG. 7 es una ilustración de cuatro subparticiones divididas en la herramienta ISP;

La FIG. 8 es una ilustración de una intrapredicción de líneas de referencia múltiple;

10

La FIG. 9 ilustra un método de selección de muestras de referencia para intrapredicción PLANAR;

La FIG. 10 ilustra otro método de selección de muestras de referencia para modo de intrapredicción PLANAR;

15

La FIG. 11 ilustra otro método más de selección de muestras de referencia para modo de intrapredicción PLANAR;

La FIG. 12 ilustra un método de selección de muestras de referencia para intrapredicción DC según una realización de la presente descripción;

20

Las FIGs. 12A-12C ilustran otros métodos de selección de muestras de referencia diferentes para el modo de intrapredicción DC según realizaciones de la presente descripción;

La FIG. 13 ilustra otro método más de selección de muestras de referencia diferentes para el modo de intrapredicción DC según realizaciones de la presente descripción;

25

La FIG. 14 ilustra un método según una realización de la presente descripción;

La FIG. 15 ilustra un codificador según la presente descripción;

30

La FIG. 16 ilustra un decodificador según la presente descripción;

La FIG. 17 es un diagrama de bloques que muestra una estructura de ejemplo de un sistema 3100 de suministro de contenido que realiza un servicio de entrega de contenido;

35

La FIG. 18 es un diagrama de bloques que muestra una estructura de un ejemplo de un dispositivo terminal.

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

40

Descripción detallada de las realizaciones

En la siguiente descripción, se hace referencia a las figuras adjuntas, que forman parte de la descripción, y que muestran, a modo de ilustración, aspectos específicos de realizaciones de la invención o aspectos específicos en los que se pueden usar las realizaciones de la presente invención. Se entiende que las realizaciones de la invención se pueden usar en otros aspectos y comprenden cambios estructurales o lógicos no representados en las figuras. Por lo tanto, la siguiente descripción detallada no debe tomarse en un sentido limitante, y el alcance de la presente invención se define mediante las reivindicaciones adjuntas.

45

Por ejemplo, se entiende que una descripción en relación con un método descrito también puede ser válida para un dispositivo o sistema correspondiente configurado para realizar el método y viceversa. Por ejemplo, si se describen uno o una pluralidad de etapas del método específicas, un dispositivo correspondiente puede incluir una o una pluralidad de unidades, por ejemplo, unidades funcionales, para realizar la una o la pluralidad de etapas del método descritas (por ejemplo, una unidad que realiza la una o la pluralidad de etapas, o una pluralidad de unidades, cada una de las cuales realiza una o más de la pluralidad de etapas), incluso si tales una o más unidades no se describen o ilustran explícitamente en las figuras. Por otro lado, por ejemplo, si un aparato específico se describe en función de una o una pluralidad de unidades, por ejemplo, unidades funcionales, un método correspondiente puede incluir una etapa para realizar la funcionalidad de la una o la pluralidad de unidades (por ejemplo, una etapa que realiza la funcionalidad de la una o la pluralidad de unidades, o una pluralidad de etapas, cada una de las cuales realiza la funcionalidad de una o más de la pluralidad de unidades), incluso si tal una o pluralidad de etapas no se describen o ilustran explícitamente en las figuras. Además, se entiende que las características de las diversas realizaciones ejemplares y/o los aspectos descritos en la presente memoria pueden combinarse entre sí, a menos que se indique específicamente lo contrario.

50

55

60

65

La codificación de vídeo generalmente se refiere al procesamiento de una secuencia de imágenes, que forman el vídeo o la secuencia de vídeo. En lugar del término "imagen", el término "trama" o "imagen" se puede usar como sinónimos en el campo de la codificación de vídeo. La codificación de vídeo (o codificación en general) comprende dos partes, codificación de vídeo y decodificación de vídeo. La codificación de vídeo se realiza en el lado de origen, lo que generalmente comprende el procesamiento (por ejemplo, mediante compresión) de las imágenes de vídeo originales para reducir la cantidad de datos requeridos para representar las imágenes de vídeo (para un almacenamiento y/o transmisión más eficiente). La decodificación de vídeo se realiza en el lado de destino y generalmente comprende el procesamiento inverso en comparación con el codificador para reconstruir las imágenes de vídeo. Las realizaciones que se refieren a la "codificación" de imágenes de vídeo (o imágenes en general) se entenderán relacionadas con la "codificación" o la "decodificación" de imágenes de vídeo o secuencias de vídeo respectivas. La combinación de la parte de codificación y la parte de decodificación también se denomina CODEC (Codificación y Decodificación).

En caso de codificación de vídeo sin pérdidas, las imágenes de vídeo originales pueden reconstruirse, es decir, las imágenes de vídeo reconstruidas tienen la misma calidad que las imágenes de vídeo originales (suponiendo que no haya pérdida de transmisión u otra pérdida de datos durante el almacenamiento o la transmisión). En caso de codificación de vídeo con pérdidas, se realiza una compresión adicional, por ejemplo, mediante cuantificación, para reducir la cantidad de datos que representan las imágenes de vídeo, que no pueden reconstruirse completamente en el decodificador, es decir, la calidad de las imágenes de vídeo reconstruidas es menor o peor en comparación con la calidad de las imágenes de vídeo originales.

Varios estándares de codificación de vídeo pertenecen al grupo de los "códecs de vídeo híbridos con pérdidas" (es decir, combinan la predicción espacial y temporal en el dominio de la muestra y la codificación por transformación 2D para aplicar la cuantificación en el dominio de la transformación). Cada imagen de una secuencia de vídeo generalmente se particiona en un conjunto de bloques que no se superponen y la codificación generalmente se realiza a nivel de bloque. En otras palabras, en el codificador, el vídeo generalmente se procesa, es decir, se codifica, en un nivel de bloque (bloque de vídeo), por ejemplo, usando predicción espacial (intraimagen) y/o predicción temporal (interimagen) para generar un bloque de predicción, restando el bloque de predicción del bloque actual (bloque actualmente procesado/a procesar) para obtener un bloque residual, transformando el bloque residual y cuantificando el bloque residual en el dominio de transformación para reducir la cantidad de datos a transmitir (compresión), mientras que en el decodificador se aplica al bloque codificado o comprimido el procesamiento inverso en comparación con el codificador para reconstruir el bloque actual para su representación. Además, el codificador duplica el bucle de procesamiento del decodificador de manera que ambos generarán predicciones idénticas (por ejemplo, intrapredicciones e interpredicciones) y/o reconstrucciones para su procesamiento, es decir, la codificación, de los bloques subsiguientes.

Se describen un sistema 10 de codificación de vídeo, un codificador 20 de vídeo y un decodificador 30 de vídeo basándose en las FIGs. 1 a 3.

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

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

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

El origen 16 de imágenes 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 un dispositivo de generación de imágenes, por ejemplo, un procesador de gráficos por ordenador para generar una imagen animada por ordenador, o cualquier tipo de 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 los mismos (por ejemplo, una imagen de realidad aumentada (AR)). El origen de imágenes puede ser cualquier tipo de memoria o almacenamiento que almacene cualquiera de las imágenes antes mencionadas.

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

5 El preprocesador 18 está configurado para recibir los datos 17 de imagen (sin procesar) y para realizar el preprocesamiento de los datos 17 de imagen para obtener una imagen 19 preprocesada o datos 19 de imagen preprocesados. 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 18 de preprocesamiento puede ser un componente opcional.

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

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

20 El dispositivo 14 de destino comprende un decodificador 30 (por ejemplo, un decodificador 30 de vídeo), y puede comprender adicionalmente, es decir, opcionalmente, una interfaz de comunicación o unidad 28 de comunicación, una posprocesador 32 (o unidad 32 de posprocesamiento) y un dispositivo 34 de visualización.

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

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

35 La interfaz 22 de comunicación puede configurarse, por ejemplo, para empaquetar los datos 21 de imagen codificados en un formato apropiado, por ejemplo, paquetes, y/o procesar los datos de imagen codificados usando cualquier tipo de codificación de transmisión o procesamiento para su transmisión a través de un enlace de comunicación o una red de comunicación.

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

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

50 El decodificador 30 está configurado para recibir los datos 21 de imagen codificados y proporcionar datos 31 de imagen decodificados o una imagen 31 decodificada (a continuación se describirán más detalles, por ejemplo, basándose en la FIG. 3 o la FIG. 5). El posprocesador 32 del dispositivo 14 de destino está configurado para posprocesar los datos 31 de imagen decodificados (también llamados datos de imagen reconstruidos), por ejemplo, la imagen 31 decodificada, para obtener datos 33 de imagen posprocesados, por ejemplo, una imagen 33 posprocesada. El posprocesamiento realizado por la unidad 32 de posprocesamiento puede comprender, por ejemplo, conversión de formato de color (por ejemplo, de YCbCr a RGB), corrección de color, recorte o remuestreo, o cualquier otro procesamiento, por ejemplo, para preparar los datos 31 de imagen decodificados para su visualización, por ejemplo, mediante el dispositivo 34 de visualización.

60 El dispositivo 34 de visualización del dispositivo 14 de destino está configurado para recibir los datos 33 de imagen posprocesados para visualizar la imagen, por ejemplo, a un usuario o espectador. El dispositivo 34 de

visualización 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 comprender, por ejemplo, pantallas de cristal líquido (LCD), pantallas de diodos orgánicos emisores de luz (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 12 de origen y el dispositivo 14 de destino como dispositivos separados, las realizaciones de los dispositivos también pueden comprender ambos o ambas funcionalidades, el dispositivo 12 de origen o la funcionalidad correspondiente y el dispositivo 14 de destino o la funcionalidad correspondiente. En tales realizaciones, el dispositivo 12 de origen o la funcionalidad correspondiente y el dispositivo 14 de destino o la funcionalidad correspondiente pueden implementarse usando el mismo hardware y/o software o mediante hardware y/o software separados o cualquier combinación de los mismos.

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

El codificador 20 (por ejemplo, un codificador 20 de vídeo) y el decodificador 30 (por ejemplo, un decodificador 30 de vídeo) o tanto el codificador 20 como el decodificador 30 puede implementarse a través de un circuito de procesamiento como se muestra en la FIG. 1B, tal como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), lógica discreta, hardware o codificación de vídeo dedicada o cualquier combinación de los mismos. El codificador 20 puede implementarse a través del circuito 46 de procesamiento para incorporar los diversos módulos como se analizó con respecto al codificador 20 de la FIG. 2 y/o cualquier otro sistema o subsistema codificador descrito en la presente memoria. El decodificador 30 puede implementarse a través del circuito 46 de procesamiento para incorporar los diversos módulos como se analizó con respecto al decodificador 30 de la FIG. 3 y/o cualquier otro sistema o subsistema decodificador descrito en la presente memoria. El circuito lógico puede configurarse para realizar las diversas operaciones que se analizan 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 soporte de almacenamiento legible por ordenador no transitorio adecuado y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta descripción. Cualquiera de los codificadores 20 de vídeo y decodificadores 30 de vídeo pueden integrarse como parte de un codificador/decodificador (CODEC) combinado en un único dispositivo, por ejemplo, como se muestra en la FIG. 1B.

El dispositivo 12 de origen y el dispositivo 14 de destino pueden comprender cualquiera de una amplia gama de dispositivos, incluido cualquier tipo de dispositivo portátil o fijo, por ejemplo, notebook u ordenadores portátiles, teléfonos móviles, teléfonos inteligentes, tabletas u ordenadores tabletas, cámaras, ordenadores de escritorio, decodificadores, televisores, dispositivos de visualización, reproductores de soportes digitales, consolas de videojuegos, dispositivos de transmisión de vídeo (tales como servidores de servicios de contenido o servidores de entrega de contenido), dispositivo receptor de emisiones, dispositivo transmisor de emisiones o similares, y puede usar ningún o cualquier tipo de sistema operativo. En algunos casos, el dispositivo 12 de origen y el dispositivo 14 de destino pueden estar equipados para comunicación inalámbrica. Por lo tanto, el dispositivo 12 de origen y el dispositivo 14 de destino pueden ser dispositivos de comunicación inalámbrica.

En algunos casos, el sistema 10 de codificación de vídeo ilustrado en la FIG. 1A es simplemente un ejemplo y las técnicas de la presente solicitud pueden aplicarse a configuraciones de codificación de vídeo (por ejemplo, codificación de vídeo o decodificación de vídeo) 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 similar. Un dispositivo de codificación de vídeo puede codificar y almacenar datos en la memoria, y/o un dispositivo de decodificación de vídeo puede recuperar y decodificar datos de la memoria. En algunos ejemplos, la codificación y decodificación la realizan dispositivos que no se comunican entre sí, sino que simplemente codifican datos en la memoria y/o recuperan y decodifican datos de la memoria.

La FIG. 1C muestra un diagrama de bloques que muestra un ejemplo de un dispositivo móvil configurado para implementar realizaciones de la presente descripción.

Para facilitar la descripción, las realizaciones de la invención se describen en la presente memoria, por ejemplo, haciendo referencia a la Codificación de Vídeo de Alta Eficiencia (HEVC) o al software de referencia de Codificación de Vídeo Versátil (VVC), estándar de codificación de vídeo de próxima generación desarrollado por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo del UIT-T (VCEG) y del Grupo de Expertos en Imágenes en Movimiento del ISO/IEC (MPEG). Un experto en la técnica comprenderá que las realizaciones de la invención no se limitan a HEVC o

VVC.

Codificador y Método de Codificación

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

La unidad 204 de cálculo residual, la unidad 206 de procesamiento de transformación, la unidad 208 de cuantificación, la unidad 260 de selección de modo pueden denominarse formadoras de una ruta de señal de reenvío del codificador 20, mientras la unidad 210 de cuantificación inversa, la unidad 212 de procesamiento de transformación inversa, la unidad 214 de reconstrucción, el búfer 216, el filtro 220 de bucle, el búfer de imágenes decodificada (DPB) 230, la unidad 244 de interpredicción y la unidad 254 de intrapredicción puede denominarse formadoras de una ruta de señal hacia atrás del codificador 20 de vídeo, en donde la ruta de señal hacia atrás del codificador 20 de vídeo corresponde a la ruta de señal del decodificador (vea el decodificador 30 en la FIG. 3). La unidad 210 de cuantificación inversa, la unidad 212 de procesamiento de transformación inversa, la unidad 214 de reconstrucción, el filtro 220 de bucle, el búfer de imágenes decodificadas (DPB) 230, la unidad 244 de interpredicción y la unidad 254 de intrapredicción pueden denominarse también formadoras del "decodificador incorporado" del codificador 20 de vídeo.

30 Imágenes y Partición de Imágenes (Imágenes y Bloques)

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

40 Una imagen (digital) es o puede ser considerada como un conjunto o matriz bidimensional de muestras con valores de intensidad. Una muestra en la matriz también puede denominarse píxel (forma abreviada de elemento de imagen) o un pel. El número de muestras en dirección horizontal y vertical (o eje) de la matriz o imagen define el tamaño y/o resolución de la imagen. Para la representación del color, generalmente se emplean tres componentes de color, es decir, la imagen puede representarse o incluir tres matrices de muestra. En espacio de color o formato RGB, una imagen comprende una matriz de muestras roja, verde y azul correspondiente. Sin embargo, en la codificación de vídeo, cada píxel generalmente se representa en un espacio de color o formato de luminancia y crominancia, por ejemplo, YCbCr, que comprende un componente de luminancia indicado por Y (a veces también se usa L en su lugar) y dos componentes de crominancia indicados por Cb y Cr. El componente Y de luminancia (o de forma abreviada, luma) representa el brillo o la intensidad del nivel de grises (por ejemplo, como en una imagen en escala de grises), mientras que los dos componentes Cb y Cr de crominancia (o de forma abreviada, croma) representan los componentes de cromaticidad o información de color. En consecuencia, una imagen en formato YCbCr comprende una matriz de muestras de luminancia de valores (Y) de muestras de luminancia y dos matrices de muestras de crominancia de valores (Cb y Cr) de crominancia. Las imágenes en formato RGB se pueden convertir o transformar en formato YCbCr y viceversa, el proceso también se denomina transformación o conversión de color. Si una imagen es monocromática, la imagen puede comprender solo una matriz de muestras de luminancia. En consecuencia, una imagen puede ser, por ejemplo, una matriz de muestras de luma en formato monocromático o una matriz de muestras de luma y dos matrices correspondientes de muestras de croma en el formato de color 4:2:0, 4:2:2 y 4:4:4.

Las realizaciones del codificador 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 203 de imágenes (generalmente no superpuestas). 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 puede configurarse para usar el mismo tamaño del bloque para todas las imágenes de una secuencia de vídeo y la cuadrícula correspondiente que define el tamaño del bloque, o para cambiar el tamaño del bloque entre imágenes, subconjuntos o grupos de imágenes, y particionar cada imagen en los bloques correspondientes.

5

En realizaciones adicionales, el codificador de vídeo 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 203 de imagen también puede denominarse bloque de imagen actual o bloque de imagen a codificar.

10 Al igual que la imagen 17, el bloque 203 de imagen nuevamente es o puede ser considerado como un conjunto o matriz 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, una matriz de muestras (por ejemplo, una matriz de luma en el caso de una imagen 17 monocromática, o una matriz de luma o croma en caso de una imagen de color) o tres matrices de muestras (por ejemplo, una matriz de luma y dos matrices de croma en el caso de una imagen 17 en color) o cualquier otro número y/o tipo de matrices 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 ser, por ejemplo, una matriz de muestras $M \times N$ (M columnas por N filas), o una matriz $M \times N$ de coeficientes de transformación.

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

25 Las realizaciones del codificador 20 de vídeo como se muestra en la FIG. 2 puede configurarse además para particionar y/o codificar la imagen usando sectores (también denominados sectores de vídeo), en donde una imagen puede particionarse o codificarse usando uno o más segmentos (generalmente no superpuestos), y cada sector puede comprender uno o más bloques (por ejemplo, CTU).

30 Las realizaciones del codificador 20 de vídeo como se muestra en la FIG. 2 puede configurarse además para particionar y/o codificar la imagen usando grupos de mosaicos (también denominados grupos de mosaicos de vídeo) y/o mosaicos (también denominados mosaicos de vídeo), en donde una imagen puede particionarse o codificarse usando uno o más grupos de mosaicos (generalmente no superpuestos), y cada grupo de mosaicos puede comprender, por ejemplo, uno o más bloques (por ejemplo, CTU) o uno o más mosaicos, en donde cada mosaico, por ejemplo, puede tener forma rectangular y puede comprender uno o más bloques (por ejemplo, CTU), por ejemplo, bloques completos o fraccionados.

35

Cálculo Residual

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

Transformación

45

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

50

55 La unidad 206 de procesamiento de transformación puede configurarse para aplicar aproximaciones enteras de DCT/DST, tales como las transformaciones especificadas para H.265/HEVC. En comparación con una transformación DCT ortogonal, tales aproximaciones de números enteros generalmente se escalan mediante un cierto factor. Para preservar la norma del bloque residual que se procesa mediante transformaciones directas e inversas, se aplican factores de escala adicionales como parte del proceso de transformación. Los factores de escala se eligen generalmente basándose en ciertas restricciones, como que los factores de escala sean una potencia de dos para las operaciones de desplazamiento, la profundidad de bits de los coeficientes de transformación, el equilibrio entre la precisión y los costes de implementación, etc. Los factores de escala específicos se especifican, por ejemplo, para la transformación inversa, por ejemplo, mediante la unidad 212 de procesamiento de transformación inversa (y la transformación inversa correspondiente, por ejemplo mediante la unidad 312 de procesamiento de transformación inversa en el decodificador 30 de vídeo) y los factores de escala correspondientes para la transformación directa, por ejemplo mediante la unidad 206 de procesamiento de transformación, en el codificador 20 pueden especificarse en consecuencia.

65

Las realizaciones del codificador 20 de vídeo (respectivamente la unidad 206 de procesamiento de transformación) pueden configurarse para generar parámetros de transformación, por ejemplo, un tipo de transformación o transformaciones por ejemplo, directamente o codificadas o comprimidas a través de la unidad 270 de codificación de entropía, para que, por ejemplo, el decodificador 30 de vídeo pueda recibir y usar los parámetros de transformación para decodificar.

Cuantificación

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

El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes 207 de transformación. Por ejemplo, un coeficiente de transformación de n bits puede redondearse hacia abajo a un coeficiente de transformación de m bits durante la cuantificación, donde 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 escalas diferentes para lograr una cuantificación más fina o más gruesa. Los tamaños de los escalones de cuantificación menores corresponden a una cuantificación más fina, mientras que los tamaños de los escalones de cuantificación más grandes corresponden a una cuantificación más gruesa. El tamaño de los escalones de cuantificación aplicables puede indicarse mediante un parámetro de cuantificación (QP). El parámetro de cuantificación puede ser, por ejemplo, un índice de un conjunto predefinido de tamaños de escalones de cuantificación aplicables. Por ejemplo, los parámetros de cuantificación pequeños pueden corresponder a una cuantificación fina (tamaños de escalones de cuantificación pequeños) y los parámetros de cuantificación grandes pueden corresponder a una cuantificación gruesa (tamaños de escalones de cuantificación grandes) o viceversa. La cuantificación puede incluir la división por un tamaño del escalón de cuantificación y una correspondiente y/o decuantificación inversa, por ejemplo, mediante la unidad 210 de cuantificación inversa, puede incluir la multiplicación por el tamaño del escalón de cuantificación. Las realizaciones según algunos estándares, por ejemplo, HEVC, pueden configurarse para usar un parámetro de cuantificación para determinar el tamaño del escalón de cuantificación. Generalmente, el tamaño del escalón de cuantificación puede calcularse basándose en un parámetro de cuantificación que usa una aproximación de punto fijo de una ecuación que incluye la división. Pueden introducirse factores de escala adicionales para la cuantificación y la decuantificación para restaurar la norma del bloque residual, que podría modificarse debido a la escala usada en la aproximación de punto fijo de la ecuación para el tamaño del escalón de cuantificación y el parámetro de cuantificación. En una implementación de ejemplo, se pueden combinar el escalado de la transformación inversa y la decuantificación. Alternativamente, pueden usarse tablas de cuantificación personalizadas y señalarse desde un codificador a un decodificador, por ejemplo, en un flujo de bits. La cuantificación es una operación con pérdida, en donde la pérdida aumenta al aumentar los tamaños de los escalones de cuantificación.

Las realizaciones del codificador 20 de vídeo (respectivamente la unidad 208 de cuantificación) pueden configurarse para generar parámetros de cuantificación (QP), por ejemplo, directamente o codificados a través de la unidad 270 de codificación de entropía, para que, por ejemplo, el decodificador 30 de vídeo pueda recibir y aplicar los parámetros de cuantificación para la decodificación.

Cuantización inversa

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

Transformación Inversa

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

Reconstrucción

La unidad 214 de reconstrucción (por ejemplo, sumador 214) está configurada para añadir el bloque 213 de transformación (es decir, el bloque 213 residual reconstruido) al bloque 265 de predicción para obtener un bloque 215 reconstruido en el dominio de la muestra, por ejemplo, añadiendo - muestra por muestra - los valores de muestra del bloque 213 residual reconstruido y los valores de muestra del bloque 265 de predicción.

Filtrado

La unidad 220 de filtro de bucle (o de forma abreviada "filtro de bucle" corto 220), está configurada para filtrar el bloque 215 reconstruido para obtener un bloque 221 filtrado, o en general, para filtrar muestras reconstruidas para obtener muestras filtradas. La unidad de filtro de bucle está, por ejemplo, configurada para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. La unidad 220 de filtro de bucle puede comprender uno o más filtros de bucle, tales como un filtro de desbloqueo, un filtro de desplazamiento adaptativo de muestras (SAO) o uno o más de otros filtros, por ejemplo, un filtro bilateral, un filtro de bucle adaptativo (ALF), unos filtros de nitidez, unos filtros de suavizado o unos filtros colaborativos, 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 220 de filtro de bucle puede implementarse como un filtro posbucle. El bloque 221 filtrado también puede denominarse bloque 221 filtrado reconstruido.

Las realizaciones del codificador 20 (respectivamente, la unidad 220 de filtro de bucle) pueden configurarse para generar parámetros de filtro de bucle (tales como información de desplazamiento adaptativa de muestra), por ejemplo, directamente o codificados a través de la unidad 270 de codificación de entropía, para que, por ejemplo, un decodificador 30 puede recibir y aplicar los mismos parámetros de filtro de bucle para la decodificación.

Búfer de Imagen Decodificada

El búfer de imágenes decodificadas (DPB) 230 puede ser una memoria que almacena imágenes de referencia, o en general datos de imagen de referencia, para codificar datos de vídeo por el codificador 20 de vídeo. El DPB 230 puede estar formado por cualquiera de una variedad de dispositivos de memoria, tales como memoria de acceso aleatorio dinámica (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 estar configurado para almacenar uno o más bloques 221 filtrados. El búfer 230 de imágenes decodificadas puede configurarse además para almacenar otros bloques previamente filtrados, por ejemplo, bloques 221 previamente reconstruidos y filtrados, de la misma imagen actual o de diferentes imágenes, por ejemplo, imágenes previamente reconstruidas, y puede proporcionar imágenes completas previamente reconstruidas, es decir, decodificadas (y los correspondientes bloques y muestras de referencia) y/o una imagen actual parcialmente reconstruida (y los correspondientes bloques y muestras de referencia), por ejemplo, para 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 se filtra mediante la unidad 220 de filtro de bucle, o cualquier otra versión procesada adicionalmente de los bloques o muestras reconstruidos.

Selección de Modo (Partición y Predicción)

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 imagen original, por ejemplo, un bloque 203 original (bloque 203 actual de la imagen actual 17), y datos de imagen reconstruidos, 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 previamente decodificadas, por ejemplo del búfer 230 de imágenes decodificadas u otros búferes (por ejemplo, búfer de línea, no mostrado). Los datos de imagen reconstruidos se usan como datos de imagen de referencia para predicción, por ejemplo, interpredicción o intrapredicción, para obtener un bloque 265 de predicción o un predictor 265. La unidad 260 de selección de modo puede configurarse para determinar o seleccionar una partición para un modo de predicción de bloque actual (incluida la no partición) y/o un modo de predicción (por ejemplo, modo de intrapredicción o interpredicción) y generar un bloque 265 de predicción correspondiente, que es usado para el cálculo del bloque 205 residual y para la reconstrucción del bloque 215 reconstruido.

Las realizaciones de la unidad 260 de selección de modo pueden configurarse para seleccionar la partición y el modo de predicción (por ejemplo, de aquellos soportados por o disponibles para la unidad 260 de selección de modo), que proporciona la mejor coincidencia o, en otras palabras, el residual mínimo (el residual mínimo significa una mejor compresión para la transmisión o almacenamiento) o una sobrecarga de señalización mínima (la sobrecarga de señalización mínima significa una mejor compresión para la transmisión o el almacenamiento), o que considera o equilibra ambos. La unidad 260 de selección de modo puede configurarse para determinar la partición y el modo de predicción basándose en la optimización de distorsión de tasa (RDO),

es decir, seleccionar el modo de predicción que proporciona una distorsión de tasa mínima. Términos como "mejor", "mínimo", "óptimo", etc. en este contexto no necesariamente se refieren a un "mejor", "mínimo", "óptimo", etc. sino que también pueden referirse al cumplimiento de una terminación o criterio de 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 262 de partición puede configurarse para particionar el bloque 203 en subbloques o particiones de bloque más pequeñas, por ejemplo, de forma iterativa usando partición de árbol cuádruple (QT), partición binario (BT) o partición de árbol triple (TT) o cualquier combinación de los mismos, y para realizar, por ejemplo, la predicción para cada una de las particiones de bloque o subbloques, 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, la partición (por ejemplo, mediante la unidad 260 de partición) y el procesamiento de predicción (mediante la unidad 244 de interpretación y la unidad 254 de intrapredicción) realizados por un codificador 20 de vídeo de ejemplo se explicarán con más detalle.

Partición

La unidad 262 de partición puede particionar (o dividir) un bloque 203 actual en particiones más pequeñas, por ejemplo, bloques más pequeños de tamaño cuadrado o rectangular. Estos bloques más pequeños (que también pueden denominarse subbloques) pueden particionarse en particiones aún más pequeñas. Esto también se denomina partición de árbol o partición de árbol jerárquico, en donde un bloque raíz, por ejemplo en el nivel 0 del árbol raíz (nivel 0 de jerarquía, profundidad 0), puede particionarse de forma recursiva, por ejemplo, particionarse en dos o más bloques de un siguiente nivel de árbol inferior, por ejemplo, nodos en el nivel 1 de árbol (nivel 1 de jerarquía, profundidad 1), en donde estos bloques pueden particionarse nuevamente en dos o más bloques de un siguiente nivel inferior, por ejemplo, nivel 2 de árbol (nivel 2 de jerarquía, profundidad 2), etc. hasta que termina la partición, por ejemplo debido a que 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 están más particionados también se denominan bloques de hojas o nodos de hojas del árbol. Un árbol que usa la partición en dos particiones se denomina árbol binario (BT), un árbol que usa la partición en tres particiones se denomina árbol ternario (TT) y un árbol que usa la partición en cuatro particiones se denomina árbol cuádruple (QT).

Como se mencionó anteriormente, el término "bloque" como se usa en la presente memoria puede ser una porción, en particular una porción cuadrada o rectangular, de una imagen. Con referencia, por ejemplo, a 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 a la 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 CTB correspondientes de muestras de croma de una imagen que tiene tres matrices de muestras, o un CTB de muestras de una imagen monocromática o una imagen que se codifica usando tres planos de color separados y estructuras de sintaxis usadas para codificar las muestras. En consecuencia, un bloque de árbol de codificación (CTB) puede ser un bloque NxN de muestras para algún valor de N, de manera que la división de un componente en CTB sea 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 matrices de muestras, o un bloque de codificación de muestras de una imagen monocroma o una imagen que se codifica usando tres planos de color separados y estructuras de sintaxis usadas para codificar las muestras. En consecuencia, un bloque de codificación (CB) puede ser un bloque MxN de muestras para algunos valores de M y N, de manera que la división de un CTB en bloques de codificación es una partición.

Según HEVC, una unidad de árbol de codificación (CTU) puede dividirse en CU usando una estructura de árbol cuádruple denominada árbol de codificación. La decisión de si codificar un área de imagen usando predicción interimagen (temporal) o intraimagen (espacial) se toma en el nivel de CU. Cada CU puede dividirse en una, dos o cuatro PU según el tipo de división de PU. Dentro de una PU, se aplica el mismo proceso de predicción y la información relevante se transmite al decodificador en función de la PU. Después de obtener el bloque residual aplicando el proceso de predicción basándose en el tipo de división de PU, una CU se puede particionar en unidades de transformación (TU) según otra estructura de árbol cuádruple similar al árbol de codificación para la CU.

Según el último estándar de codificación de vídeo actualmente en desarrollo, que se denomina Codificación de Vídeo Versátil (VVC), se usa, por ejemplo, una partición combinada de árbol cuádruple y árbol binario

(QTBT) para particionar un bloque de codificación. En la estructura de bloques QTBT, una CU puede tener una forma cuadrada o rectangular. Por ejemplo, una unidad de árbol de codificación (CTU) se particiona primero mediante una estructura de árbol cuádruple. Los nodos de hoja de árbol cuádruple se particionan además mediante una estructura de árbol binario o árbol ternario (o triple). Los nodos de hoja de árbol de partición se denominan unidades de codificación (CU), y esa segmentación se usa para el procesamiento de predicción y transformación sin ninguna otra partición. Esto significa que la CU, PU y TU tienen el mismo tamaño del bloque en la estructura de bloques de codificación QTBT. En paralelo, también se puede usar la partición múltiple, por ejemplo, la partición de árbol triple junto con la estructura de bloques QTBT.

En un ejemplo, la unidad 260 de selección de modo del codificador 20 de vídeo puede configurarse para realizar cualquier combinación de las técnicas de partición descritas en la presente memoria.

Como se describió anteriormente, el codificador 20 está configurado para determinar o seleccionar el modo de predicción mejor u óptimo de un conjunto de modos de predicción (por ejemplo, predeterminados). 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 soporte) 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 soporte) y el modo planar, o modos direccionales, por ejemplo, como se define por VVC.

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

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

Interpredicción

El conjunto de (o posibles) modos de interpredicción depende de las imágenes de referencia disponibles (es decir, imágenes anteriores al menos parcialmente decodificadas, por ejemplo, almacenadas en DBP 230) y otros parámetros de interpredicción, por ejemplo, si se usa 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 para buscar un bloque de referencia que coincida mejor, y/o por ejemplo, si se aplica la interpolación de píxeles, por ejemplo, interpolación de soporte/semipel y/o de 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 244 de interpredicción puede incluir una unidad de estimación de movimiento (ME) y una unidad de compensación de movimiento (MC) (no mostrada en la FIG. 2). La unidad de estimación de movimiento puede configurarse para recibir u obtener el bloque 203 de imagen (bloque 203 de imagen actual de la imagen 17 actual) y una imagen 231 decodificada, o al menos uno o una pluralidad de bloques previamente reconstruidos, por ejemplo, bloques reconstruidos de uno o una pluralidad de otras/diferentes imágenes 231 previamente decodificadas, para estimación de movimiento. Por ejemplo, una secuencia de vídeo puede comprender la imagen actual y las imágenes 231 previamente decodificadas, o en otras palabras, la imagen actual y las imágenes 231 previamente decodificadas pueden ser parte o formar una secuencia de imágenes que forman una secuencia de vídeo.

El codificador 20 puede, por ejemplo, configurarse para seleccionar un bloque de referencia de una pluralidad de bloques de referencia de las mismas 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 para 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 para realizar una interpredicción basándose en o usando el parámetro de interpredicción

para obtener un bloque 265 de interpredicción. La compensación de movimiento, realizada por la unidad de compensación de movimiento, puede implicar obtener o generar el bloque de predicción basándose en el vector de movimiento/bloque determinado por la estimación de movimiento, posiblemente realizando interpolaciones con precisión de subpíxel. 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 usarse para codificar un bloque de imagen. Al recibir el vector de movimiento para la PU del bloque de imagen actual, la unidad de compensación de movimiento puede ubicar 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 con los bloques y el segmento de vídeo para su uso por el decodificador 30 de vídeo en la decodificación de los bloques de imágenes del segmento de vídeo. Además o como alternativa a los segmentos y los respectivos elementos de sintaxis, se pueden generar o usar grupos de mosaicos y/o mosaicos y los respectivos elementos de sintaxis.

Codificación de Entropía

La unidad 270 de codificación de entropía está configurada para aplicar 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, una codificación aritmética binaria adaptable al contexto (CABAC), una codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una codificación de entropía de partición de intervalo de probabilidad (PIPE) u otra técnica o metodología de codificación de entropía) o derivación (no compresión) en los coeficientes 209 residuales cuantificados, los parámetros de interpredicción, un parámetro de intrapredicción, los parámetros de filtro de bucle y/u otros elementos de sintaxis para obtener datos 21 de imagen codificados que pueden ser generados por la salida 272, por ejemplo, en la forma de un flujo 21 de bits codificados, para que, por ejemplo, el decodificador 30 de vídeo pueda recibir y usar los parámetros para decodificar. El flujo 21 de bits codificados puede transmitirse al decodificador 30 de vídeo, o almacenarse en una memoria para su posterior transmisión o recuperación mediante el decodificador 30 de vídeo.

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

Decodificador y Método de Decodificación

La FIG. 3 muestra un ejemplo de un decodificador 30 de vídeo que está configurado para implementar las técnicas de esta presente solicitud. El decodificador 30 de vídeo está configurado para recibir datos 21 de imagen codificados (por ejemplo, un flujo 21 de bits codificado), por ejemplo, codificados por el codificador 20, para obtener una imagen 331 decodificada. Los datos de imagen codificados o flujo de bits comprenden información para decodificar los datos de imagen codificados, por ejemplo, datos que representan bloques de imágenes de un segmento de vídeo codificado (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 304 de decodificación de entropía, una unidad 310 de cuantificación inversa, una unidad 312 de procesamiento de transformación inversa, una unidad 314 de reconstrucción (por ejemplo, un sumador 314), un filtro 320 de bucle, un búfer de imágenes decodificadas (DBP) 330, una unidad 360 de aplicación de modo, una unidad 344 de interpredicción y una unidad 354 de intrapredicción. La unidad 344 de interpredicción puede ser o incluir una unidad de compensación de movimiento. El decodificador 30 de vídeo puede, en algunos ejemplos, realizar una etapa de decodificación generalmente recíproca a la etapa de codificación descrita con respecto al codificador 100 de vídeo de la FIG. 2.

Como se explicó con respecto al codificador 20, la unidad 210 de cuantificación inversa, la unidad 212 de procesamiento de transformación inversa, la unidad 214 de reconstrucción, el filtro 220 de bucle, el búfer de imágenes decodificadas (DPB) 230, la unidad 344 de interpredicción y la unidad 354 de intrapredicción también se denominan formadores de "decodificador integrado" del codificador 20 de vídeo. En consecuencia, la unidad 310 de cuantificación inversa puede tener una función idéntica a la unidad 110 de cuantificación inversa, la unidad 312 de procesamiento de transformación inversa puede tener una función idéntica a la unidad 212 de procesamiento de transformación inversa, la unidad 314 de reconstrucción puede tener una función idéntica a la unidad 214 de reconstrucción, el filtro 320 de bucle puede tener una función idéntica al filtro 220 de bucle, y el búfer 330 de imágenes decodificadas puede tener una función idéntica al búfer 230 de imagen decodificada. Por lo tanto, las explicaciones proporcionadas para las respectivas unidades y funciones del codificador 20 de

vídeo se aplican en consecuencia a las respectivas unidades y funciones del decodificador 30 de vídeo.

Decodificación de Entropía

- 5 La unidad 304 de decodificación de entropía está configurada para analizar el flujo 21 de bits (o, en general, los datos 21 de imagen codificados) y realizar, por ejemplo, una decodificación de entropía de los datos 21 de imagen codificados para obtener, por ejemplo, coeficientes 309 cuantificados y/o parámetros de codificación decodificados (no mostrados en la FIG. 3), por ejemplo, alguno o todos los parámetros de interpredicción (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 como se describe con respecto a la unidad 270 de codificación de entropía del codificador 20. La unidad 304 de decodificación de entropía está configurada además para proporcionar parámetros de interpredicción, parámetro de intrapredicción y/u otros elementos de sintaxis a la unidad 360 de aplicación de modo y otros parámetros a otras unidades del decodificador 30. El decodificador 30 de vídeo puede recibir los elementos de sintaxis en el nivel de segmento de vídeo y/o el nivel de bloque de vídeo. Además o como alternativa a los segmentos y los respectivos elementos de sintaxis, se pueden recibir o usar grupos de mosaicos y/o mosaicos y los respectivos elementos de sintaxis.

Cuantización Inversa

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

Transformación Inversa

- 35 La unidad 312 de procesamiento de transformación inversa puede configurarse para recibir coeficientes 311 decuantificados, también denominados coeficientes 311 de transformación, y para aplicar una transformación a los coeficientes 311 decuantificados con el fin de obtener bloques 213 residuales reconstruidos en la muestra dominio. El bloque 213 residual reconstruido también puede denominarse bloque 313 de transformación. La transformación puede ser una transformación inversa, por ejemplo, una DCT inversa, una DST inversa, una transformación de números enteros inversa o un proceso de transformación inversa conceptualmente similar. La unidad 312 de procesamiento de transformación inversa puede configurarse además para recibir parámetros de transformación o información correspondiente de los datos 21 de imagen codificados (por ejemplo, mediante análisis y/o decodificación, por ejemplo mediante la unidad 304 de decodificación de entropía) para determinar la transformación que se aplicará a los coeficientes 311 decuantificados.

Reconstrucción

- La unidad 314 de reconstrucción (por ejemplo, el sumador 314) está configurada para añadir el bloque 313 de transformación inversa al bloque 365 de predicción para obtener un bloque 315 reconstruido en el dominio de la muestra, por ejemplo, sumando los valores de muestra del bloque 313 residual reconstruido y los valores de muestra del bloque 365 de predicción.

Filtrado

- 55 La unidad 320 de filtro de bucle (ya sea en el bucle de codificación o después del bucle de codificación) está configurada para filtrar el bloque 315 reconstruido para obtener un bloque 321 filtrado, por ejemplo, para suavizar las transiciones de píxeles o de otro modo mejorar la calidad del vídeo. La unidad 320 de filtro de bucle puede comprender uno o más filtros de bucle, tal como un filtro de desbloqueo, un filtro de desplazamiento adaptativo de muestra (SAO) o uno o más de otros filtros, por ejemplo, un filtro de bucle adaptativo (ALF), un filtro de nitidez, unos filtros de suavizado o unos filtros colaborativos 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 320 de filtro de bucle puede implementarse como un filtro posbucle.

Búfer de Imagen Decodificada

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

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

Predicción

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

- 20 Cuando el segmento de vídeo se codifica como un segmento intracodificado (I), la unidad 354 de intrapredicción de la unidad 360 de aplicación de modo está configurada para generar el bloque 365 de predicción para un bloque de imagen del segmento de vídeo actual basándose en un modo de intrapredicción señalado y datos de bloques previamente decodificados de la imagen actual. Cuando la imagen de vídeo se codifica como un segmento intercodificado (es decir, B o P), la unidad 344 de interpredicción (por ejemplo, la unidad de compensación de movimiento) de la unidad 360 de aplicación de modo está configurada para producir bloques
- 25 365 de predicción para un bloque de vídeo del segmento de vídeo actual basándose en los vectores de movimiento y otros elementos de sintaxis recibidos de la unidad 304 de decodificación de entropía. 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 30 de vídeo puede construir las listas de tramas de referencia, Lista 0 y Lista 1, usando técnicas de construcción predeterminadas basándose en
- 30 imágenes de referencia almacenadas en DPB 330. Se puede aplicar lo mismo o similar cuando se usan grupos de mosaicos (por ejemplo, grupos de mosaicos de vídeo) y/o mosaicos (por ejemplo, mosaicos de vídeo) además o alternativamente a segmentos (por ejemplo, segmentos de vídeo), por ejemplo, un vídeo puede codificarse usando grupos de mosaicos y/o mosaicos I, P o B.

- 35 La unidad 360 de aplicación de modo está configurada para determinar la información de predicción para un bloque de vídeo del segmento de vídeo actual analizando los vectores de movimiento o información relacionada y otros elementos de sintaxis, y usa la información de predicción para producir los bloques de predicción para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad 360 de aplicación de modo usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo,
- 40 intrapredicción o interpredicción) usado para codificar los bloques de vídeo del segmento de vídeo, un tipo de segmento 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 vídeo intercodificado del segmento, estado de interpredicción para cada bloque de vídeo intercodificado del segmento, y otra información para decodificar los bloques de vídeo en el segmento de vídeo
- 45 actual. Se puede aplicar lo mismo o similar cuando se usan grupos de mosaicos (por ejemplo, grupos de mosaicos de vídeo) y/o mosaicos (por ejemplo, mosaicos de vídeo) además o alternativamente a segmentos (por ejemplo, segmentos de vídeo), por ejemplo, un vídeo puede codificarse usando grupos de mosaicos y/o mosaicos I, P o B.

- 50 El decodificador 30 de vídeo como se muestra en la FIG. 3 puede configurarse para particionar y/o decodificar la imagen usando sectores (también denominados segmentos de vídeo), en donde una imagen puede particionarse o decodificarse usando uno o más segmentos (generalmente no superpuestos), y cada segmento puede comprender uno o más bloques (por ejemplo, CTU).

- 55 El decodificador 30 de vídeo como se muestra en la FIG. 3 puede configurarse además para particionar y/o decodificar la imagen usando grupos de mosaicos (también denominados grupos de mosaicos de vídeo) y/o mosaicos (también denominados mosaicos de vídeo), en donde una imagen puede particionarse o decodificarse usando uno o más grupos de mosaicos (generalmente no superpuestos), y cada grupo de mosaicos puede comprender, por ejemplo, uno o más bloques (por ejemplo, CTU) o uno o más mosaicos, en
- 60 donde cada mosaico, por ejemplo, puede tener forma rectangular y puede comprender uno o más bloques (por ejemplo, CTU), por ejemplo, bloques completos o fraccionados.

- 65 Se pueden usar otras variaciones del decodificador 30 de vídeo para decodificar los datos 21 de imagen codificados. Por ejemplo, el decodificador 30 puede producir el flujo de vídeo de salida sin la unidad 320 de filtrado de bucle. Por ejemplo, un codificador 30 no basado en transformación puede cuantificar de forma

inversa la señal residual directamente sin la unidad 312 de procesamiento de transformación inversa para ciertos bloques o tramas. En otra implementación, el decodificador 30 de vídeo puede tener la unidad 310 de cuantificación inversa y la unidad 312 de procesamiento de transformación inversa combinadas en una sola unidad.

5

Debe entenderse que, en el codificador 20 y el decodificador 30, un resultado de procesamiento de una etapa actual puede procesarse adicionalmente y luego enviarse a la siguiente etapa. Por ejemplo, después del filtrado por interpolación, la derivación del vector de movimiento o el filtrado de bucle, se puede realizar una operación adicional, tal como Recorte o desplazamiento, en el resultado del procesamiento del filtrado por interpolación, derivación del vector de movimiento o filtrado de bucle.

10

Cabe señalar que se pueden aplicar operaciones adicionales a los vectores de movimiento derivados del bloque actual (incluidos, pero sin limitarse a, vectores de movimiento del punto de control de modo afín, vectores de movimiento de subbloque en afín, plano, modos ATMVP, vectores de movimiento temporal, etc.). Por ejemplo, el valor del vector de movimiento está restringido a un intervalo predefinido según su bit representativo. Si el bit que representa el vector de movimiento es bitDepth, entonces el intervalo es $-2^{(\text{bitDepth}-1)} \sim 2^{(\text{bitDepth}-1)}$, donde "^" significa exponenciación. Por ejemplo, si bitdepth se establece en 16, el intervalo es -32768 ~ 32767; si bitdepth se establece en 18, el intervalo es -131072-131071. Por ejemplo, el valor del vector de movimiento derivado (por ejemplo, los MV de cuatro subbloques de 4x4 dentro de un bloque de 8x8) está restringido de manera que la diferencia máxima entre partes enteras de los MV de cuatro subbloques de 4x4 no sea más de N píxeles, tal como no más de 1 píxel. Aquí se proporcionan dos métodos para restringir el vector de movimiento según el bitDepth.

15

20

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

25

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

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

30

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

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

donde mvx es un componente horizontal de un vector de movimiento de un bloque o un subbloque de imagen, mv y es un componente vertical de un vector de movimiento de un bloque o un subbloque de imagen, 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 igual que el resultado aplicando la fórmula (1) y (2).

35

40

$$ux = (mv_{px} + mv_{dx} + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (5)$$

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

45

$$uy = (mv_{py} + mv_{dy} + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}} \quad (7)$$

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

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

55

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

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

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 un subbloque de imagen; x, y y z corresponden respectivamente a tres valores de entrada del proceso de recorte MV, y la definición de la función Clip3 es la siguiente:

60

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

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

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

El procesador 430 está implementado mediante hardware y software. El procesador 430 puede implementarse como uno o más chips de CPU, núcleos (por ejemplo, como un procesador multinúcleo), FPGA, ASIC y DSP. El procesador 430 está en comunicación con el puerto de entrada 410, la unidad 420 de recepción, la unidad 440 de transmisión, los puertos 450 de salida y la memoria 460. El procesador 430 comprende un módulo 470 de codificación. El módulo 470 de codificación implementa las realizaciones descritas anteriormente. Por ejemplo, el módulo 470 de codificación implementa, procesa, prepara o proporciona las diversas operaciones de codificación. La inclusión del módulo 470 de codificación por lo tanto proporciona una mejora sustancial a la funcionalidad del dispositivo 400 de codificación de vídeo y efectúa una transformación del dispositivo 400 de codificación de vídeo a un estado diferente. De manera alternativa, el módulo 470 de codificación se implementa como instrucciones almacenadas en la memoria 460 y ejecutadas por el procesador 430.

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

La FIG. 5 es un diagrama de bloques simplificado de un aparato 500 que se puede usar como uno o ambos dispositivo 12 de origen y dispositivo 14 de destino 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 en la actualidad o desarrollada en el futuro. Aunque las implementaciones descritas se pueden practicar con un solo procesador como se muestra, por ejemplo, el procesador 502, se pueden lograr ventajas en velocidad y eficiencia usando más de un procesador.

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

El aparato 500 también puede incluir uno o más dispositivos de salida, tales 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 puede operar para detectar entradas táctiles. La pantalla 518 se puede acoplar al procesador 502 a través del bus 512.

Aunque representado aquí como un solo bus, el bus 512 del aparato 500 puede estar compuesto por múltiples buses. Además, el almacenamiento 514 secundario se puede acoplar directamente 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, tal como una tarjeta de memoria, o múltiples unidades, tales como múltiples tarjetas de memoria. De este modo, el aparato 500 puede implementarse en una amplia variedad de configuraciones.

Los datos 21 de imagen codificados contienen información sobre cómo se realiza una predicción de bloque en el codificador (unidad 254) y el decodificador (unidad 354). La sintaxis de los datos de imagen codificados puede comprender los siguientes elementos (Tabla 1):

5

Tabla 1. Elementos de sintaxis ejemplares relacionados con la intrapredicción de un bloque

...	
si(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
si((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx[x0][y0]	ae(v)
si(intra_luma_ref_idx[x0][y0] == 0 && (cbWidth <= MaxTbSizeY cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag[x0][y0]	ae(v)
si(intra_subpartitions_mode_flag[x0][y0] == 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag[x0][y0]	ae(v)
si(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
si(intra_luma_mpm_flag[x0][y0]) {	
si(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_not_planar_flag[x0][y0]	ae(v)
si(intra_luma_not_planar_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
} de lo contrario	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
si(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
...	

intra_luma_ref_idx[x0][y0] especifica el índice de línea de referencia de intrapredicción IntraLumaRefLineIdx[x][y] para x = x0..x0 + cbWidth - 1 e y = y0..y0 + cbHeight - 1 como se especifica en la Tabla 2. Cuando intra_luma_ref_idx[x0][y0] no está presente, se infiere que es igual a 0.

10

Tabla 2. Especificación de IntraLumaRefLineIdx[x][y] basándose en intra_luma_ref_idx[x0][y0].

intra_luma_ref_idx[x0][y0]	IntraLumaRefLineIdx[x][y]
	x = x0..x0 + cbWidth - 1 y = y0..y0 + cbHeight - 1
0	0
1	1
2	3

IntraLumaRefLineIdx se usa para determinar el valor del índice de línea de referencia de intrapredicción (refIdx):

15

$$\text{refIdx} = (\text{cIdx} == 0) ? \text{IntraLumaRefLineIdx}[\text{xTbCmp}][\text{yTbCmp}] : 0,$$

donde cIdx es una variable que indica el componente de luma o croma de un bloque predicho. cIdx igual a 0 indica componente de luma y cIdx igual a 1 o 2 indica componentes de croma.

En la descripción dada adicionalmente, cbWidth corresponde al ancho del bloque, cbHeight corresponde a la altura del bloque, MaxTbSizeY corresponde al valor umbral, que podría definirse de la siguiente manera:

5 MaxTbLog2SizeY = 6,

MaxTbSizeY = 1 << MaxTbLog2SizeY

El bloque predicho también se denomina bloque de codificación.

10 intra_subpartitions_mode_flag[x0][y0] igual a 1 especifica que la unidad de intracodificación actual se particiona en NumIntraSubPartitions [x0][y0] subparticiones de bloques de transformación rectangulares.

15 intra_subpartitions_mode_flag[x0][y0] igual a 0 especifica que la unidad de codificación intra actual no está particionada en subparticiones de bloques de transformación rectangulares.

Cuando intra_subpartitions_mode_flag[x0][y0] no está presente, se infiere que es igual a 0. intra_subpartitions_split_flag[x0][y0] especifica si el tipo de división intrasubparticiones es horizontal o vertical. Cuando intra_subpartitions_split_flag[x0][y0] no está presente, se infiere de la siguiente manera:

20 – Si cbHeight es mayor que MaxTbSizeY, se infiere que intra_subpartitions_split_flag[x0][y0] es igual a 0.

– De lo contrario (cbWidth es mayor que MaxTbSizeY), se infiere que intra_subpartitions_split_flag[x0][y0] es igual a 1.

25 La variable IntraSubPartitionsSplitType especifica el tipo de división usada para el bloque de codificación luma actual como se ilustra en la Tabla 3. IntraSubPartitionsSplitType se deriva de la siguiente manera:

30 – Si intra_subpartitions_mode_flag[x0][y0] es igual a 0, IntraSubPartitionsSplitType se establece en 0.

– De lo contrario, el IntraSubPartitionsSplitType se establece en 1 + intra_subpartitions_split_flag[x0][y0].

Tabla 3 - Asociación de nombres con IntraSubPartitionsSplitType

IntraSubPartitionsSplitType	Nombre de IntraSubPartitionsSplitType
0	ISP_NO_SPLIT
1	ISP_HOR_SPLIT
2	ISP_VER_SPLIT

35 La variable NumIntraSubPartitions especifica el número de subparticiones de bloque de transformación en las que se divide un bloque de codificación intraluma. NumIntraSubPartitions se deriva de la siguiente manera:

– Si IntraSubPartitionsSplitType es igual a ISP_NO_SPLIT, NumIntraSubPartitions se establece en 1.

40 – De lo contrario, si una de las siguientes condiciones es verdadera, NumIntraSubPartitions se establece igual a 2:

– cbWidth es igual a 4 y cbHeight es igual a 8,

45 – cbWidth es igual a 8 y cbHeight es igual a 4,

– De lo contrario, NumIntraSubPartitions se establece en 4.

50 Los elementos de sintaxis intra_luma_mpm_flag[x0][y0], intra_luma_not_planar_flag[x0][y0], intra_luma_mpm_idx[x0][y0] e intra_luma_mpm_remainder[x0][y0] especifican el modo de intrapredicción para muestras de luma. Los índices de matriz x0, y0 especifican la ubicación (x0 , y0) de la muestra de luma superior izquierda del bloque de codificación considerado en relación con la muestra de luma superior izquierda de la imagen. Cuando intra_luma_mpm_flag[x0][y0] es igual a 1, el modo de intrapredicción se infiere a partir de una unidad de codificación intrapredicha vecina.

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

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

Cuando $\text{intra_luma_not_planar_flag}[x_{Cb}][y_{Cb}]$ es igual a 0, el modo de intrapredicción del bloque ($\text{IntraPredModeY}[x_{Cb}][y_{Cb}]$) se establece igual a INTRA_PLANAR .

- 5 $\text{intra_chroma_pred_mode}[x_0][y_0]$ especifica el modo de intrapredicción para muestras de croma (componente de croma del bloque). Los índices de matriz x_0, y_0 especifican la ubicación (x_0, y_0) de la muestra de luma superior izquierda del bloque de codificación considerado en relación con la muestra de luma superior izquierda de la imagen.
- 10 Los detalles de la herramienta de intrasubpartición (ISP) se pueden encontrar en JVET-M0102. A continuación se da una breve descripción de esta herramienta:

La herramienta ISP divide los bloques intrapredichos de luma vertical u horizontalmente en 2 o 4 subparticiones, dependiendo de las dimensiones del tamaño del bloque, como se muestra en la Tabla 4. FIGs. 6 y 7 muestran ejemplos de las dos posibilidades. Todas las subparticiones cumplen la condición de tener al menos 16 muestras.

Tabla 4: Número de subparticiones según el tamaño del bloque

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

- 20 Para cada una de estas subparticiones, se genera una señal residual decodificando por entropía los coeficientes enviados por el codificador y luego su cuantización inversa y transformación inversa. Luego, la subpartición se intrapredice y finalmente las muestras reconstruidas correspondientes se obtienen añadiendo la señal residual a la señal de predicción. Por tanto, los valores reconstruidos de cada subpartición estarán disponibles para generar la predicción de la siguiente, que repetirá el proceso y así sucesivamente. Todas las subparticiones comparten el mismo intramodo.

Basándose en el intramodo y la división utilizada, se usan dos clases diferentes de órdenes de procesamiento, que se denominan orden normal y orden invertido. En el orden normal, la primera subpartición que se procesará es la que contiene la muestra superior izquierda de la CU y luego continúa hacia abajo (división horizontal) o hacia la derecha (división vertical). Como resultado, las muestras de referencia usadas para generar las señales de predicción de subparticiones solo están ubicadas en los lados izquierdo y superior de las líneas. Por otro lado, el orden de procesamiento inverso comienza con la subpartición que contiene la muestra inferior izquierda de la CU y continúa hacia arriba o comienza con la subpartición que contiene la muestra superior derecha de la CU y continúa hacia la izquierda.

- 35 Generalmente, el proceso de intrapredicción podría describirse como sigue.

Las entradas a este proceso son:

- 40 – una ubicación de muestra (x_{TbCmp}, y_{TbCmp}) que especifica la muestra superior izquierda del bloque de transformación actual en relación con la muestra superior izquierda de la imagen actual,
- una variable predModelIntra que especifica el modo de intrapredicción,
- 45 – una variable n_{TbW} que especifica el ancho del bloque de transformación,
- una variable n_{TbH} que especifica la altura del bloque de transformación,
- una variable n_{CbW} que especifica el ancho del bloque de codificación,
- 50 – una variable n_{CbH} que especifica la altura del bloque de codificación,
- una variable $cldx$ que especifica el componente de color del bloque actual.

- 55 Las salidas de este proceso son las muestras predichas $\text{predSamples}[x][y]$, con $x = 0..n_{TbW} - 1$, $y = 0..n_{TbH} - 1$.

Las variables refW y refH se derivan de la siguiente manera:

- Si IntraSubPartitionsSplitType es igual a ISP_NO_SPLIT o cldx no es igual a 0, se aplica lo siguiente:

$$\text{refW} = \text{nTbW} * 2$$

$$\text{refH} = \text{nTbH} * 2$$

- De lo contrario (IntraSubPartitionsSplitType no es igual a ISP_NO_SPLIT o cldx es igual a 0), se aplica lo siguiente:

$$\text{refW} = \text{nCbW} * 2$$

$$\text{refH} = \text{nCbH} * 2$$

La variable refIdx que especifica el índice de línea de referencia de intrapredicción se deriva de la siguiente manera:

$$\text{refIdx} = (\text{cldx} == 0) ? \text{IntraLumaRefLineIdx}[\text{xTbCmp}][\text{yTbCmp}] : 0$$

Para la generación de las muestras de referencia $p[x][y]$ con $x = -1 - \text{refIdx}$, $y = -1 - \text{refIdx} - \text{refH} - 1$ y $x = -\text{refIdx} - \text{refW} - 1$, $y = -1 - \text{refIdx}$, se aplican las siguientes etapas ordenadas:

1. El proceso de marcado de disponibilidad de la muestra de referencia se invoca con la ubicación de la muestra (xTbCmp , yTbCmp), el índice de línea de referencia de intrapredicción refIdx, el ancho de la muestra de referencia refW, la altura de la muestra de referencia refH, el índice del componente de color cldx como entradas y las muestras de referencia refUnfilt[x][y] con $x = -1 - \text{refIdx}$, $y = -1 - \text{refIdx} - \text{refH} - 1$ y $x = -\text{refIdx} - \text{refW} - 1$, $y = -1 - \text{refIdx}$ como salida.

2. Cuando al menos una muestra refUnfilt[x][y] con $x = -1 - \text{refIdx}$, $y = -1 - \text{refIdx} - \text{refH} - 1$ y $x = -\text{refIdx} - \text{refW} - 1$, $y = -1 - \text{refIdx}$ está marcada como "no disponible para intrapredicción", el proceso de sustitución de la muestra de referencia se invoca con el índice de línea de referencia de intrapredicción refIdx, el ancho de la muestra de referencia refW, la altura de la muestra de referencia refH, las muestras de referencia refUnfilt[x][y] con $x = -1 - \text{refIdx}$, $y = -1 - \text{refIdx} - \text{refH} - 1$ y $x = -\text{refIdx} - \text{refW} - 1$, $y = -1 - \text{refIdx}$, y el índice del componente de color cldx como entradas, y las muestras de referencia modificadas refUnfilt[x][y] con $x = -1 - \text{refIdx}$, $y = -1 - \text{refIdx} - \text{refH} - 1$ y $x = -\text{refIdx} - \text{refW} - 1$, $y = -1 - \text{refIdx}$ como salida.

3. El proceso de filtrado de muestras de referencia se invoca con el índice de línea de referencia de intrapredicción refIdx, el ancho del bloque de transformación nTbW y la altura nTbH, el ancho de la muestra de referencia refW, la altura de la muestra de referencia refH, las muestras sin filtrar refUnfilt[x][y] con $x = -1 - \text{refIdx}$, $y = -1 - \text{refIdx} - \text{refH} - 1$ y $x = -\text{refIdx} - \text{refW} - 1$, $y = -1 - \text{refIdx}$, y el índice del componente de color cldx como entradas, y las muestras de referencia $p[x][y]$ con $x = -1 - \text{refIdx}$, $y = -1 - \text{refIdx} - \text{refH} - 1$ y $x = -\text{refIdx} - \text{refW} - 1$, $y = -1 - \text{refIdx}$ como salida.

El proceso de predicción intramuestra según predModelIntra se aplica de la siguiente manera:

– Si predModelIntra es igual a INTRA_PLANAR, el proceso de modo de intrapredicción correspondiente se invoca con el ancho del bloque de transformación nTbW, la altura del bloque de transformación nTbH y la matriz de muestras de referencia p como entradas, y la salida es la matriz de muestras predichas predSamples.

– De lo contrario, si predModelIntra es igual a INTRA_DC, el proceso de modo de intrapredicción correspondiente se invoca con el ancho del bloque de transformación nTbW, la altura del bloque de transformación nTbH, el índice de línea de referencia de predicción refIdx, y la matriz de muestras de referencia p como entradas, y la salida es la matriz de muestras predicha predSamples.

– De lo contrario, si predModelIntra es igual a INTRA_LT_CCLM, INTRA_L_CCLM o INTRA_T_CCLM, el proceso del modo de intrapredicción correspondiente se invoca con el modo de intrapredicción predModelIntra, la ubicación de la muestra (xTbC , yTbC) establecida en (xTbCmp , yTbCmp), el ancho del bloque de transformación nTbW y la altura nTbH, y la matriz de muestras de referencia p como entradas, y la salida es la matriz de muestras predicha predSamples.

– De lo contrario, el proceso del modo de intrapredicción correspondiente se invoca con el modo de intrapredicción predModelIntra, el índice de línea de referencia de intrapredicción refIdx, el ancho del bloque de transformación nTbW, la altura del bloque de transformación nTbH, el ancho de la muestra de referencia refW, la altura de la muestra de referencia refH, el ancho del bloque de codificación nCbW y la altura nCbH, el índice del componente de color cldx y la matriz de muestras de referencia p como entradas, y el modo de intrapredicción modificado predModelIntra y la matriz de muestras predicha predSamples como salidas.

Cuando todas las condiciones siguientes son verdaderas, el proceso de filtrado de muestras de predicción dependiente de la posición se invoca con el modo de intrapredicción `predModelIntra`, el ancho del bloque de transformación `nTbW`, la altura del bloque de transformación `nTbH`, las muestras predichas `predSamples[x][y]`, con $x = 0..nTbW - 1$, $y = 0..nTbH - 1$, el ancho de la muestra de referencia `refW`, la altura de la muestra de referencia `refH`, las muestras de referencia `p[x][y]`, con $x = -1$, $y = -1..refH - 1$ y $x = 0..refW - 1$, $y = -1$, y el índice del componente de color `cldx` como entradas, y la salida es la matriz de muestras predicha modificada `predSamples`:

- 10 – `IntraSubPartitionsSplitType` es igual a `ISP_NO_SPLIT` o `cldx` no es igual a 0
- `refldx` es igual a 0 o `cldx` no es igual a 0
- Todas las siguientes condiciones son verdaderas:
- 15 – `predModelIntra` es igual a `INTRA_PLANAR`
- `predModelIntra` es igual a `INTRA_DC`
- 20 – `predModelIntra` es igual a `INTRA_ANGULAR18`
- `predModelIntra` es igual a `INTRA_ANGULAR50`
- `predModelIntra` es menor o igual que `INTRA_ANGULAR10`
- 25 – `predModelIntra` es mayor o igual que `INTRA_ANGULAR58`

La FIG. 8 muestra posiciones de líneas de muestras de referencia que se usan para la intrapredicción de un bloque. Dependiendo del índice de línea de referencia ("Línea de referencia"), las muestras del bloque predicho ("Unidad de bloque") pueden predecirse usando muestras de referencia que no son adyacentes al bloque predicho, es decir, cuando el índice de línea de referencia no es igual a cero.

El modo de intrapredicción planar se especifica en el Borrado de VVC de la siguiente manera:

- 35 Las entradas a este proceso son:
- una variable `nTbW` que especifica el ancho del bloque de transformación,
- una variable `nTbH` que especifica la altura del bloque de transformación,
- 40 – las muestras vecinas `p[x][y]`, con $x = -1$, $y = -1..nTbH$ y $x = 0..nTbW$, $y = -1$.

Las salidas de este proceso son las muestras predichas `predSamples[x][y]`, con $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

- 45 Las variables `nW` y `nH` se derivan de la siguiente manera:

$$nW = \text{Max}(nTbW, 2)$$

$$50 \quad nH = \text{Max}(nTbH, 2)$$

Los valores de las muestras de predicción `predSamples[x][y]`, con $x = 0..nTbW - 1$ e $y = 0..nTbH - 1$, se derivan de la siguiente manera:

$$55 \quad \text{predV}[x][y] = ((nH - 1 - y) * p[x][y - 1] + (y + 1) * p[x][y]) \ll \text{Log2}(nW)$$

$$\text{predH}[x][y] = ((nW - 1 - x) * p[x - 1][y] + (x + 1) * p[x][y]) \ll \text{Log2}(nH)$$

$$\text{predSamples}[x][y] = (\text{predV}[x][y] + \text{predH}[x][y] + nW * nH) \gg (\text{Log2}(nW) + \text{Log2}(nH) + 1)$$

- 60 El proceso de intrapredicción DC se especifica en el borrador de VVC de la siguiente manera:

Las entradas a este proceso son:

- una variable nTbW que especifica el ancho del bloque de transformación,
- una variable nTbH que especifica la altura del bloque de transformación,
- 5 – las muestras vecinas $p[x][y]$, con $x = -1, y = -1..nTbH - 1$ y $x = 0..nTbW - 1, y = -1$.

Las salidas de este proceso son las muestras predichas $predSamples[x][y]$, con $x = 0..nTbW - 1, y = 0..nTbH - 1$. Los valores de las muestras de predicción $predSamples[x][y]$, con $x = 0..nTbW - 1, y = 0..nTbH - 1$, se derivan mediante las siguientes etapas ordenadas:

- 10 1. Una variable dcVal se deriva de la siguiente manera:

- Cuando nTbW es igual a nTbH:

$$15 \quad dcVal = \left(\sum_{x'=-1}^{nTbW-1} p[x'][-1] + \sum_{y'=-1}^{nTbH-1} p[-1][y'] + nTbW \right) \gg (\text{Log2}(nTbW) + 1)$$

- Cuando nTbW es mayor que nTbH:

$$20 \quad dcVal = \left(\sum_{x'=-1}^{nTbW-1} p[x'][-1] + (nTbW \gg 1) \right) \gg \text{Log2}(nTbW)$$

- Cuando nTbW es menor que nTbH:

$$25 \quad dcVal = \left(\sum_{y'=-1}^{nTbH-1} p[-1][y'] + (nTbH \gg 1) \right) \gg \text{Log2}(nTbH)$$

2. Las muestras de predicción $predSamples[x][y]$ se derivan de la siguiente manera:

$predSamples[x][y] = dcVal$, con $x = 0..nTbW-1, y = 0..nTbH-1$

- 30 Los métodos del estado de la técnica usan la intrapredicción direccional solo cuando el índice de línea de referencia es distinto de cero, es decir, el modo de intrapredicción DC y el modo de intrapredicción PLANAR están desactivados en este caso.

- 35 Otro caso en el que no se realiza el modo de intrapredicción DC es cuando se usa la intrasubpartición (ISP). La versión del borrador de VVC especifica una codificación diferente del modo de intrapredicción. La selección de la secuencia de elementos de sintaxis señalados en el flujo de bits depende de que las herramientas particulares se activen o desactiven. En concreto, el indicador mpm se señala solo cuando el indicador "intra_luma_ref_idx" y el indicador "intra_subpartitions_mode_flag" son iguales a cero (consulte la Tabla 5).

- 40 Tabla 5. Codificación del índice intramodo según lo especificado por la sintaxis de la unidad de codificación en el borrador del texto de especificación de VVC versión 4 revisión 7.

si(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
si((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx[x0][y0]	ae(v)
si (intra_luma_ref_idx[x0][y0] == 0 && (cbWidth <= MaxTbSizeY cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag[x0][y0]	ae(v)
si(intra_subpartitions_mode_flag[x0][y0] == 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag[x0][y0]	ae(v)
si(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
si(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)

de lo contrario	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
si(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)

PDPC son las siglas de combinación de predicción dependiente de la posición. Este proceso podría describirse de la siguiente manera:

5 Las entradas a este proceso son:

- el modo de intrapredicción predModelIntra,
- una variable nTbW que especifica el ancho del bloque de transformación,
- 10 – una variable nTbH que especifica la altura del bloque de transformación,
- una variable refW que especifica el ancho de las muestras de referencia,
- 15 – una variable refH que especifica la altura de las muestras de referencia,
- las muestras predichas predSamples[x][y], con $x = 0..nTbW - 1$, $y = 0..nTbH - 1$,
- las muestras vecinas p[x][y], con $x = -1$, $y = -1..refH - 1$ y $x = 0..refW - 1$, $y = -1$,
- 20 – una variable cldx que especifica el componente de color del bloque actual.

Las salidas de este proceso son las muestras predichas modificadas predSamples[x][y], con

25 $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

Dependiendo del valor de cldx, la función clip1Cmp se establece de la siguiente manera:

- Si cldx es igual a 0, clip1Cmp se establece en Clip1_y.
- 30 – De lo contrario, clip1Cmp se establece en Clip1_c.

La variable nScale se establece en $((\text{Log2}(nTbW) + \text{Log2}(nTbH) - 2) \gg 2)$.

35 Las matrices de muestra de referencia mainRef[x] y sideRef[y], con $x = 0..refW - 1$ e $y = 0..refH - 1$ se derivan de la siguiente manera:

mainRef[x] = p[x][-1]

40 sideRef[y] = p[-1][y]

Las variables refL[x][y], refT[x][y], wT[y], wL[x] y wTL[x][y] con $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ se derivan de la siguiente manera:

45 – Si predModelIntra es igual a INTRA_PLANAR o INTRA_DC, se aplica lo siguiente:

refL[x][y] = p[-1][y]

refT[x][y] = p[x][-1]

50 $wT[y] = 32 \gg ((y \ll 1) \gg nScale)$

$wL[x] = 32 \gg ((x \ll 1) \gg nScale)$

55 $wTL[x][y] = (\text{predModelIntra} == \text{INTRA_DC}) ? ((wL[x] \gg 4) + (wT[y] \gg 4)) : 0$

– De lo contrario, si predModelIntra es igual a INTRA_ANGULAR18 o INTRA_ANGULAR50, se aplica lo

siguiente:

$refL[x][y] = p[-1][y]$

5 $refT[x][y] = p[x][-1]$

$wT[y] = (predModelIntra == INTRA_ANGULAR18) ? 32 >> ((y << 1) >> nScale) : 0$

$wL[x] = (predModelIntra == INTRA_ANGULAR50) ? 32 >> ((x << 1) >> nScale) : 0$

10 $wTL[x][y] = (predModelIntra == INTRA_ANGULAR18) ? wT[y] : wL[x]$

– De lo contrario, si `predModelIntra` es igual a `INTRA_ANGULAR2` o `INTRA_ANGULAR66`, se aplica lo siguiente:

15 $refL[x][y] = p[-1][x+y+1]$

$refT[x][y] = p[x+y+1][-1]$

20 $wT[y] = (32 >> 1) >> ((y << 1) >> nScale)$

$wL[x] = (32 >> 1) >> ((x << 1) >> nScale)$

$wTL[x][y] = 0$

25 – De lo contrario, si `predModelIntra` es menor o igual que `INTRA_ANGULAR10`, se aplican las siguientes etapas ordenadas:

30 1. Las variables `dXPos[y]`, `dXFrac[y]`, `dXInt[y]` y `dX[x][y]` se derivan de la siguiente manera usando `invAngle` como se especifica en la cláusula 8.4.4.2.7 dependiendo del `intraPredMode`:

$dXPos[y] = ((y+1) * invAngle + 2) >> 2$

$dXFrac[y] = dXPos[y] \& 63$

35 $dXInt[y] = dXPos[y] >> 6$

$dX[x][y] = x + dXInt[y]$

40 2. Las variables `refL[x][y]`, `refT[x][y]`, `wT[y]`, `wL[x]` y `wTL[x][y]` se derivan de la siguiente manera:

$refL[x][y] = 0$

$refT[x][y] = (dX[x][y] < refW - 1) ? mainRef[dX[x][y] + (dXFrac[y] >> 5)] : 0$

45 $wT[y] = (dX[x][y] < refW - 1) ? 32 >> ((y << 1) >> nScale) : 0$

$wL[x] = 0$

50 $wTL[x][y] = 0$

– De lo contrario, si `predModelIntra` es mayor o igual que `INTRA_ANGULAR58`, se aplican las siguientes etapas ordenadas:

55 1. Las variables `dYPos[x]`, `dYFrac[x]`, `dYInt[x]` y `dY[x][y]` se derivan de la siguiente manera usando `invAngle` como se especifica en la cláusula 8.4.4.2.7 dependiendo del `intraPredMode`:

$dYPos[x] = ((x+1) * invAngle + 2) >> 2$

60 $dYFrac[x] = dYPos[x] \& 63$

$dYInt[x] = dYPos[x] >> 6$

$dY[x][y] = y + dYInt[x]$

2. Las variables $\text{refL}[x][y]$, $\text{refT}[x][y]$, $\text{wT}[y]$, $\text{wL}[x]$ y $\text{wTL}[x][y]$ se derivan de la siguiente manera:

5 $\text{refL}[x][y] = (\text{dY}[x][y] < \text{refH} - 1) ? \text{sideRef}[\text{dY}[x][y]] + (\text{dYFrac}[x] \gg 5) : 0$
 $\text{refT}[x][y] = 0$
 $\text{wT}[y] = 0$
10 $\text{wL}[x] = (\text{dY}[x][y] < \text{refH} - 1) ? 32 \gg ((x \ll 1) \gg \text{nScale}) : 0$
 $\text{wTL}[x][y] = 0$

15 – De lo contrario, $\text{refL}[x][y]$, $\text{refT}[x][y]$, $\text{wT}[y]$, $\text{wL}[x]$ y $\text{wTL}[x][y]$ se establecen todas igual a 0.

Los valores de las muestras predichas modificadas $\text{predSamples}[x][y]$, con $x = 0..n\text{TbW} - 1$ e $y = 0..n\text{TbH} - 1$, se derivan de la siguiente manera:

$\text{predSamples}[x][y] = \text{clip1Comp}((\text{refL}[x][y] * \text{wL}[x] + \text{refT}[x][y] * \text{wT}[y] -$
 $\text{p}[-1][y] * \text{wTL}[x][y] +$
 $(64 - \text{wL}[x] - \text{wT}[y] + \text{wTL}[x][y]) * \text{predSamples}[x][y] \div 32) \gg 6)$

20 Actualmente, se mantienen varias versiones de la lista MPM en el software VVC y en el borrador de especificación. Tampoco es deseable un diseño de hardware que use varias listas MPM. Sin embargo, si se activan los modos de intrapredicción para los casos cuando actualmente están desactivados, se producirían dos combinaciones de indicadores intra que producirían bloques intrapredichos similares. Evidentemente, este
25 diseño es redundante y para resolver este problema se debe modificar la parte de señalización o de intrapredicción.

El alcance de la invención es la modificación de los procesos de intrapredicción PLANAR y DC para los casos donde las muestras de referencia no son adyacentes al bloque que se predice o que un bloque a predecir se divide de nuevo (por ejemplo, usando ISP). Las modificaciones incluyen no solo el proceso de determinación
30 de muestras de predicción, sino también el filtrado de muestras de referencia y PDPC.

El objetivo de tal modificación es la posibilidad de tener un mecanismo de señalización de intramodo unificado para todas las herramientas de intrapredicción.

35 La invención propone modificar el procesamiento de muestras de referencia e introducir conmutación condicional de PDPC en los casos en que se usan modos de intrapredicción DC o PLANAR y cuando las muestras de referencia no son adyacentes al bloque que se está prediciendo o cuando un bloque predicho es sujeto a una nueva división (es decir, en el caso de ISP). El procesamiento incluye la selección de las muestras
40 de referencia de las muestras reconstruidas en el área superior e izquierda (como se muestra en la FIG.6) y su filtrado.

La presente invención tiene como objetivo proporcionar una posibilidad de tener un único mecanismo de señalización unificado para codificar el modo de intrapredicción. En la Tabla 6 se da un método de señalización unificada ejemplar de codificación de modo de intrapredicción.

Tabla 6. Codificación unificada ejemplar del índice intramodo.

si($\text{treeType} == \text{SINGLE_TREE} \parallel \text{treeType} == \text{DUAL_TREE_LUMA}$) {	
si($(y0 \% \text{CtbSizeY}) > 0$)	
$\text{intra_luma_ref_idx}[x0][y0]$	ae(v)
si($\text{intra_luma_ref_idx}[x0][y0] == 0 \ \&\&$ $(\text{cbWidth} \leq \text{MaxTbSizeY} \parallel \text{cbHeight} \leq \text{MaxTbSizeY}) \ \&\&$ $(\text{cbWidth} * \text{cbHeight} > \text{MinTbSizeY} * \text{MinTbSizeY})$)	
$\text{intra_subpartitions_mode_flag}[x0][y0]$	ae(v)
si($\text{intra_subpartitions_mode_flag}[x0][y0] == 1 \ \&\&$ $\text{cbWidth} \leq \text{MaxTbSizeY} \ \&\& \text{cbHeight} \leq \text{MaxTbSizeY}$)	

intra_subpartitions_split_flag[x0][y0]	ae(v)
intra_luma_mpm_flag[x0][y0]	ae(v)
si(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
de lo contrario	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
si(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)

La diferencia entre los métodos de codificación mostrados en la Tabla 5 y la Tabla 6 es que la señalización condicional de intra_luma_mpm_flag se elimina en la sintaxis ejemplar propuesta.

- 5 De la especificación se podría observar que el modo de intrapredicción PLANAR no puede funcionar cuando el índice de línea de referencia es distinto de cero. La FIG. 9 describe cómo esto podría manejarse en esta figura, el caso en el que se usa la 3ª línea de referencia. Las muestras de referencia que se seleccionan para el modo de intrapredicción PLANAR se representan mediante cuadrados rellenos con sombreado. Suponiendo que la posición de la primera muestra superior izquierda (indicada como "A") sea igual a (0, 0), la intrapredicción PLANAR podría modificarse de la siguiente manera:

Las entradas a este proceso son:

- una variable refIdx que especifica el índice de línea de referencia de intrapredicción,
- una variable nTbW que especifica el ancho del bloque de transformación,
- una variable nTbH que especifica la altura del bloque de transformación,
- las muestras de referencia p[x][y], con $x = -1 - \text{refIdx}$, $y = -1 - \text{refIdx} \dots n\text{TbH}$ y $x = -1 - \text{refIdx} \dots n\text{TbW}$, $y = -1 - \text{refIdx}$.

Las salidas de este proceso son las muestras predichas predSamples[x][y], con $x = 0 \dots n\text{TbW} - 1$, $y = 0 \dots n\text{TbH} - 1$.

Las variables nW y nH se derivan de la siguiente manera:

$$nW = \text{Max}(nTbW, 2)$$

$$nH = \text{Max}(nTbH, 2)$$

Los valores de las muestras de predicción predSamples[x][y], con $x = 0 \dots n\text{TbW} - 1$ e $y = 0 \dots n\text{TbH} - 1$, se derivan de la siguiente manera:

$$\text{predV}[x][y] = ((nH - 1 - y) * p[x][-1 - \text{refIdx}] + (y + 1) * p[-1 - \text{refIdx}][nTbH]) \ll \text{Log2}(nW)$$

$$\text{predH}[x][y] = ((nW - 1 - x) * p[-1 - \text{refIdx}][y] + (x + 1) * p[nTbW][-1 - \text{refIdx}]) \ll \text{Log2}(nH)$$

$$\text{predSamples}[x][y] = (\text{predV}[x][y] + \text{predH}[x][y] + nW * nH) \gg (\text{Log2}(nW) + \text{Log2}(nH) + 1)$$

La FIG. 10 describe otro método para seleccionar muestras de referencia de la línea de referencia para el modo de intrapredicción PLANAR. Este método podría describirse de la siguiente manera. Las entradas a este proceso son:

- una variable refIdx que especifica el índice de línea de referencia de intrapredicción,
- una variable nTbW que especifica el ancho del bloque de transformación,
- una variable nTbH que especifica la altura del bloque de transformación,

– las muestras vecinas $p[x][y]$, con $x = -1-\text{refldx}$, $y = -1-\text{refldx}..\text{nTbH}-\text{refldx}$ y $x = -\text{refldx}..\text{nTbW}-\text{refldx}$, $y = -1-\text{refldx}$.

5 Las salidas de este proceso son las muestras predichas $\text{predSamples}[x][y]$, con $x = 0..\text{nTbW} - 1$, $y = 0..\text{nTbH} - 1$.

Las variables nW y nH se derivan de la siguiente manera:

10 $nW = \text{Max}(nTbW, 2)$
 $nH = \text{Max}(nTbH, 2)$

Los valores de las muestras de predicción $\text{predSamples}[x][y]$, con $x = 0..\text{nTbW} - 1$ e $y = 0..\text{nTbH} - 1$, se derivan de la siguiente manera:

15 $\text{predV}[x][y] = ((nH - 1 - y) * p[x-\text{refldx}][y-\text{refldx}] + (y + 1) * p[x-\text{refldx}][nTbH - \text{refldx}]) \ll \text{Log2}(nW)$
 $\text{predH}[x][y] = ((nW - 1 - x) * p[x-\text{refldx}][y-\text{refldx}] + (x + 1) * p[nTbW - \text{refldx}][y-\text{refldx}]) \ll \text{Log2}(nH)$
 $\text{predSamples}[x][y] = (\text{predV}[x][y] + \text{predH}[x][y] + nW * nH) \gg (\text{Log2}(nW) + \text{Log2}(nH) + 1)$

20 Una alternativa no reivindicada consiste en aplicar un desplazamiento especial para las dos muestras cuyos valores se usan en bipredicción como valor de muestra la línea inferior y valor de muestra del lado derecho, respectivamente. En la FIG. 11 se puede observar que las muestras de referencia se seleccionan de la línea de referencia con un espacio. El texto de especificación para esta alternativa podría ser el siguiente:

25 Las entradas a este proceso son:

– una variable refldx que especifica el índice de línea de referencia de intrapredicción,
 – una variable $nTbW$ que especifica el ancho del bloque de transformación,
 30 – una variable $nTbH$ que especifica la altura del bloque de transformación,
 – las muestras vecinas $p[x][y]$, con $x = -1-\text{refldx}$, $y = -1-\text{refldx}..\text{nTbH}-\text{refldx}$ y $x = -\text{refldx}..\text{nTbW}-\text{refldx}$, $y = -1-\text{refldx}$.

35 Las salidas de este proceso son las muestras predichas $\text{predSamples}[x][y]$, con $x = 0..\text{nTbW} - 1$, $y = 0..\text{nTbH} - 1$.

Las variables nW y nH se derivan de la siguiente manera:

40 $nW = \text{Max}(nTbW, 2)$
 $nH = \text{Max}(nTbH, 2)$

45 Los valores de las muestras de predicción $\text{predSamples}[x][y]$, con $x = 0..\text{nTbW} - 1$ e $y = 0..\text{nTbH} - 1$, se derivan de la siguiente manera:

$\text{predV}[x][y] = ((nH - 1 - y) * p[x-\text{refldx}][y-\text{refldx}] + (y + 1) * p[x-\text{refldx}][nTbH - \text{refldx}]) \ll \text{Log2}(nW)$
 $\text{predH}[x][y] = ((nW - 1 - x) * p[x-\text{refldx}][y-\text{refldx}] + (x + 1) * p[nTbW - \text{refldx}][y-\text{refldx}]) \ll \text{Log2}(nH)$
 50 $\text{predSamples}[x][y] = (\text{predV}[x][y] + \text{predH}[x][y] + nW * nH) \gg (\text{Log2}(nW) + \text{Log2}(nH) + 1)$

El proceso de intrapredicción DC tampoco se pudo invocar cuando el índice de línea de referencia es distinto de cero.

55 Las realizaciones de la invención mostradas en las FIGs. 12-12C y FIG. 13 describen cómo realizar

intrapredicción usando el modo DC cuando el índice de línea de referencia es distinto de cero. El mecanismo de muestreo en este caso es similar a los descritos anteriormente para el modo de intrapredicción PLANAR:

Las entradas a este proceso son:

- 5
- una variable nTbW que especifica el ancho del bloque de transformación,
 - una variable nTbH que especifica la altura del bloque de transformación,
- 10
- una variable refIdx que especifica el índice de línea de referencia de intrapredicción,
 - las muestras vecinas $p[x][y]$, con $x = -1 - \text{refIdx}$, $y = -1 - \text{refIdx}..n\text{TbH} - \text{refIdx} - 1$ y $x = -\text{refIdx}..n\text{TbW} - \text{refIdx} - 1$, $y = -1 - \text{refIdx}$.

15 Las salidas de este proceso son las muestras predichas $\text{predSamples}[x][y]$, con $x = 0..n\text{TbW} - 1$, $y = 0..n\text{TbH} - 1$. Los valores de las muestras de predicción $\text{predSamples}[x][y]$, con $x = 0..n\text{TbW} - 1$, $y = 0..n\text{TbH} - 1$, se derivan mediante las siguientes etapas ordenadas:

1. Una variable dcVal se deriva de la siguiente manera:

- 20
- Cuando nTbW es igual a nTbH:

$$\text{dcVal} = \left(\sum_{x'=0}^{n\text{TbW}-1} p[x' - \text{refIdx}][-1 - \text{refIdx}] + \sum_{y'=0}^{n\text{TbH}-1} p[-1 - \text{refIdx}][y' - \text{refIdx}] + n\text{TbW} \right) \gg (\text{Log2}(n\text{TbW}) + 1)$$

- 25
- Cuando nTbW es mayor que nTbH:

$$\text{dcVal} = \left(\sum_{x'=0}^{n\text{TbW}-1} p[x' - \text{refIdx}][-1 - \text{refIdx}] + (n\text{TbW} \gg 1) \right) \gg \text{Log2}(n\text{TbW})$$

- Cuando nTbW es menor que nTbH:

30

$$\text{dcVal} = \left(\sum_{y'=0}^{n\text{TbH}-1} p[-1 - \text{refIdx}][y' - \text{refIdx}] + (n\text{TbH} \gg 1) \right) \gg \text{Log2}(n\text{TbH})$$

2. Las muestras de predicción $\text{predSamples}[x][y]$ se derivan de la siguiente manera:

35 $\text{predSamples}[x][y] = \text{dcVal}$, con $x = 0..n\text{TbW} - 1$, $y = 0..n\text{TbH} - 1$

La etapa 1 descrita anteriormente comprende la etapa de selección de muestras de muestra de referencia de la línea y/o columna de muestras de referencia (tal como una línea de muestra de referencia superior y/o una columna de muestra de referencia izquierda). Para este propósito, se introducen dos índices:

40 $x' = 0..n\text{TbW}$

y

45 $y' = 0..n\text{TbH}$.

La selección de muestras de referencia de la línea de muestra de referencia superior se realiza usando el índice x' y refIdx de la siguiente manera:

50 $p[x' - \text{refIdx}][-1 - \text{refIdx}]$.

La selección de muestras de referencia de la columna de muestras de referencia izquierda se realiza usando el índice y' y refIdx de la siguiente manera:

55 $p[-1 - \text{refIdx}][y' - \text{refIdx}]$

El resultado del proceso de selección son las muestras de referencia seleccionadas que se usan además para determinar el valor de la variable dcVal (valor DC) como se describe anteriormente en la etapa 1.

60 El etapa 2 de intrapredicción del bloque usando el modo de intrapredicción DC descrito anteriormente usa el valor calculado de la variable dcVal (valor DC) para obtener los valores de las muestras predichas

(predSamples), es decir, muestras de un bloque a predecir.

La FIG. 12A es similar a la FIG. 12 porque el valor de W es igual al valor de H. Sin embargo, la FIG. 12A solo muestra tres líneas de referencia, es decir, tres filas de muestras de referencia y tres columnas de muestras de referencia, respectivamente. Estas líneas de referencia están encima de la parte superior y a la izquierda del bloque a predecir.

Las FIGs. 12B y 12C son similares a la FIG. 12A, sin embargo, ilustran un caso donde $W > H$ (FIG. 12B) y $W < H$ (FIG. 12C). Es decir, en ambos casos el bloque a predecir tiene forma rectangular.

Como se muestra en las FIGs. 12-12C y 13, las muestras de referencia usadas para predecir el bloque de transformación son un subconjunto de muestras de referencia de la línea de referencia. Se debe repetir nuevamente que se debe entender que la línea de referencia incluye no solo una fila de muestras de referencia, sino también una columna de muestras de referencia. Una fila de muestras de referencia es un ejemplo de línea de referencia. Una columna de unas muestras de referencia es otro ejemplo de una línea de referencia. Las muestras de referencia usadas para predecir el bloque de transformación como se muestra en las FIGs. 12 - 12C y 13 son contiguos y no hay espacio entre ellos dentro de la línea de referencia, para que formen un intervalo contiguo o una distancia contigua. Cada una de las muestras de referencia tiene horizontal o verticalmente adyacentes a las muestras vecinas que también son para intrapredicción.

Las entradas al modo de intrapredicción INTRA_DC son:

- una variable nTbW que especifica el ancho del bloque de transformación,
- una variable nTbH que especifica la altura del bloque de transformación,
- una variable refIdx que especifica el índice de línea de referencia de intrapredicción,
- las muestras vecinas $p[x][y]$, con $x = -1 - \text{refIdx}$, $y = 0..nTbH - 1$ y $x = 0..nTbW - 1$, $y = -1 - \text{refIdx}$.

Las salidas de este proceso son las muestras predichas $\text{predSamples}[x][y]$, con $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

Como los ejemplos mostrados en las figuras 12 y 12A, el bloque es cuadrado. El ancho del bloque de transformación (nTbW) es igual a la altura del bloque de transformación (nTbH). En esta situación, una variable dcVal se deriva de la siguiente manera:

- Cuando nTbW es igual a nTbH:

$$\text{dcVal} = \left(\sum_{x'=0}^{nTbW-1} p[x'][-1 - \text{refIdx}] + \sum_{y'=0}^{nTbH-1} p[-1 - \text{refIdx}][y'] + nTbW \right) \gg (\text{Log2}(nTbW) + 1)$$

El valor del índice de línea de referencia de intrapredicción es 3 en el ejemplo de la FIG. 12, y es 2 en el ejemplo de la FIG. 12A. Tomemos el ejemplo de la FIG. 12A, el valor de refIdx es 2 y nTbW es igual a nTbH. Por lo tanto, se seleccionan la 3ª fila superior de muestras de referencia y la 3ª columna izquierda de muestras de referencia. La 3ª fila superior de muestras de referencia y la 3ª columna izquierda de muestras de referencia son las líneas de referencia indexadas por el valor del índice de línea de referencia de intrapredicción.

Como los ejemplos mostrados en las FIGs. 12B y 12C, el bloque no es cuadrado. El ancho del bloque de transformación (nTbW) no es igual a la altura del bloque de transformación (nTbH). En esta situación, una variable dcVal se deriva de la siguiente manera:

- Cuando nTbW es mayor que nTbH (Fig. 12B):

$$\text{dcVal} = \left(\sum_{x'=0}^{nTbW-1} p[x'][-1 - \text{refIdx}] + (nTbW \gg 1) \right) \gg \text{Log2}(nTbW)$$

- Cuando nTbW es menor que nTbH (Fig. 12C):

$$\text{dcVal} = \left(\sum_{y'=0}^{nTbH-1} p[-1 - \text{refIdx}][y'] + (nTbH \gg 1) \right) \gg \text{Log2}(nTbH)$$

Luego, los valores de las muestras de predicción $\text{predSamples}[x][y]$ se derivan de la siguiente manera:

$\text{predSamples}[x][y] = \text{dcVal}$, con $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

El valor del índice de línea de referencia de intrapredicción es 3 en el ejemplo de la FIG. 12B y 12C. Tomemos el ejemplo de la FIG. 12B, el valor de redIdx es 2 y nTbW es mayor que nTbH . Por lo tanto, solo se selecciona la 3ª fila superior de muestras de referencia. La 3ª fila superior de muestras de referencia es la línea de referencia indexada por el valor del índice de línea de referencia intrapredicción. El número de muestras de referencia corresponde al ancho del bloque de transformación. La muestra de referencia más a la izquierda (es decir, $p[0][-\text{idx}-1]$) de la línea de referencia se desplaza horizontalmente a la posición (es decir, A) de la muestra predicha superior izquierda dentro de la fila superior de muestras de referencia del bloque de transformación. El desplazamiento es la diferencia entre el índice vertical (es decir, $[-\text{idx}-1]$) de la muestra de referencia más a la izquierda y la posición vertical (es decir, A) de la muestra predicha superior izquierda de la columna izquierda de muestras de referencia del bloque de transformación. El desplazamiento está representado por el primer índice de la matriz p , que es igual a 0. La posición horizontal de la muestra de referencia más a la izquierda (es decir, $p[0][-\text{idx}-1]$) de la línea de referencia es igual a la posición horizontal de la muestra predicha superior izquierda (es decir, A) del bloque de transformación, y ambos son iguales a 0.

Tomemos el ejemplo de la FIG. 12C, el valor de redIdx es 2 y nTbH es mayor que nTbW . Por lo tanto, solo se selecciona la 3ª columna izquierda de muestras de referencia. La 3ª columna izquierda de muestras de referencia es la línea de referencia indexada por el valor del índice de línea de referencia intrapredicción. El número de muestras de referencia corresponde a la altura del bloque de transformación. La muestra de referencia más superior (es decir, $p[-\text{idx}-1][0]$) de la línea de referencia se desplaza verticalmente a la posición (es decir, A) de la muestra predicha superior izquierda dentro de la columna izquierda de muestras de referencia del bloque de transformación. El desplazamiento es la diferencia entre el índice horizontal (es decir, $[-\text{idx}-1]$) de la muestra de referencia más superior y la posición horizontal (es decir, A) de la muestra predicha superior izquierda de la fila superior de muestras de referencia del bloque de transformación. El desplazamiento está representado por el segundo índice de la matriz p , que es igual al valor del índice de línea de referencia intrapredicción. La posición vertical de la muestra de referencia más superior (es decir, $p[-\text{idx}-1][0]$) de la línea de referencia es igual a la posición vertical de la muestra predicha superior izquierda (es decir, A) del bloque de transformación, y ambos son iguales a 0.

Cualquier otro modo de intrapredicción (incluido el modo de intrapredicción DC) comprende las siguientes etapas:

La etapa 1 es la selección de la línea de referencia y la columna de referencia basándose en el índice de línea de referencia (refIdx). Este índice determina el desplazamiento entre el límite izquierdo o superior del bloque predicho y la columna o fila de muestras reconstruidas de los bloques vecinos que se usan además para obtener los valores de las muestras predichas.

La etapa 2 consiste en la determinación de si se aplica un filtro de muestra de referencia a la línea y columna seleccionadas de muestras de referencia. Estas condiciones comprenden el modo de intrapredicción, índice de color (cldx), valores de elementos de sintaxis que se relacionan con el proceso de intrapredicción (por ejemplo, mostrados en la Tabla 1), así como valores de variables derivadas de los valores de estos elementos, dimensión de un bloque a predecir. En caso cuando se realice ISP, estas condiciones comprenden adicionalmente el tamaño de un bloque que se dividió en subbloques a los que se aplica el proceso de intrapredicción. Según el resultado de comprobar si se cumplen las condiciones, un filtro de muestra de referencia procesa las muestras de la columna o línea de muestras de referencia seleccionadas. Este filtro suele ser un filtro FIR (respuesta de impulso finito) que tiene 3 coeficientes definidos como $[1 \ 2 \ 1] / 4$.

La etapa 3 consiste en la selección de las muestras de referencia de la columna o línea de muestras de referencia seleccionada. Esta selección consiste en la determinación del intervalo de índices que indican qué muestras pertenecientes a la línea o columna seleccionada de muestras de referencia se usarían posteriormente para derivar los valores de las muestras predichas.

La etapa 4 consiste en derivar muestras predichas a partir de las muestras de referencia seleccionadas determinadas en la etapa 3. Esta derivación depende del modo y puede comprender un submuestreo adicional, la aplicación de filtrado de interpolación o el cálculo de una expresión lineal a partir de las muestras de referencia, donde las muestras de referencia que se usan para obtener el valor de una muestra predicha se determinan dependiendo de la posición de la muestra predicha dentro de un bloque predicho.

Otra alternativa no reivindicada desactiva el suavizado de muestras de referencia intra cuando las muestras de referencia no son adyacentes al bloque que se predice, es decir, cuando el índice de la línea referenciada de muestras de referencia (señalada por "intra_luma_ref_idx") no es igual a cero. Otra alternativa no reivindicada consiste en variar el filtro de muestras de referencia para el modo de intrapredicción PLANAR con respecto al índice de la línea de referenciada de muestras de referencia.

En las tablas 7 y 8 se da una selección de filtro ejemplar.

Tabla 7. Selección de filtro ejemplar con respecto al índice de la línea de muestras de referencia referenciada

Distancia al bloque predicho (muestras)	Selección de filtro de muestra de referencia
0	[1 2 1] si el bloque predicho contiene no menos de 256 muestras, [1] de lo contrario
Otros valores	[1]

Tabla 8. Selección de filtro ejemplar con respecto al índice de la línea de muestras de referencia referenciada

Distancia al bloque predicho (muestras)	Selección de filtro de muestra de referencia
0	[1 2 1] si el bloque predicho contiene no menos de 256 muestras, [1] de lo contrario
1	[1 2 1] si el bloque predicho contiene no menos de 128 muestras, [1] de lo contrario
2	[1 0 2 0 1] si el bloque predicho contiene no menos de 256 muestras, [1] de lo contrario
3	[1 0 2 0 1] si el bloque predicho contiene no menos de 128 muestras, [1 2 1] de lo contrario

5

Otra alternativa no reivindicada de la invención permite usar los modos de intrapredicción PLANAR y DC para la intrapredicción de subparticiones cuando el indicador de intrasubpartición (ISP) está establecido en 1. Se propone no realizar la operación PDPC cuando el indicador ISP está establecido en 1.

10 - Para el caso cuando el modo ISP está activado, una realización ejemplar de cómo procesar DC intra es la siguiente: calcular el valor de DC para la muestra de referencia de todo el bloque, y

- usar el valor de DC predicho como un intrapredicador para todos los subbloques que pertenecen al bloque que se está prediciendo.

15

Otra alternativa no reivindicada describe el caso cuando los componentes de croma y luma de una imagen codificada comparten la misma partición espacial en bloques (es decir, cuando se selecciona el modo de partición SINGLE_TREE). En este caso se consideran dos casos de aplicación condicional de PDPC.

20 En el primer caso de la alternativa no reivindicada cuando el indicador ISP se establece en 1, un bloque con muestras predichas de componente de luma se divide en subbloques luma, pero el bloque croma cúbicado (es decir, un bloque cúbicado con muestras predichas de componentes de croma) no se divide. En este caso, PDPC no se aplica a los subbloques de luma, pero PDPC se aplica al bloque de croma.

25 En el segundo caso de la alternativa no reivindicada cuando el índice de línea de referencia de intrapredicción es distinto de cero, PDPC no se aplica al bloque de luma predicho, pero se aplica al bloque de croma cúbicado y para la predicción del bloque de croma el índice de línea de referencia de intrapredicción no se señala pero se infiere que es cero.

30 La FIG. 14 ilustra un diagrama de flujo de un método de intrapredicción de modo de intrapredicción DC para un bloque, que comprende una etapa 1601 de obtener un valor de un índice de línea de referencia de intra predicción de un bloque de transformación; y una etapa 1603 de obtener muestras de predicción del bloque de transformación basándose en muestras de referencia de una línea de referencia, en donde la línea de referencia está indexada por el valor del índice de línea de referencia de intrapredicción.

35

Antes de la etapa 1603, el método puede comprender una etapa 1602 de seleccionar una línea de muestras de referencia basándose en el valor del índice de línea de referencia de intrapredicción, como se muestran en los ejemplos en las FIGs. 9-13.

40 La FIG. 15 ilustra un codificador 20 según la presente descripción, comprendiendo el codificador 20: una unidad 2001 de obtención configurada para obtener un valor de un índice de línea de referencia de intrapredicción de un bloque de transformación; y una unidad 2003 de predicción configurada para obtener muestras de predicción del bloque de transformación basándose en muestras de referencia de una línea de referencia, en donde la línea de referencia está indexada por el valor del índice de línea de referencia de intrapredicción.

45

El codificador 20 puede incluir una unidad 2002 de selección (no mostrada en la FIG. 15), configurada para

seleccionar la línea de muestras de referencia basándose en el valor del índice de línea de referencia de intrapredicción, como los ejemplos mostrados en las FIGs. 9-13.

La FIG. 16 ilustra un decodificador 30 según la presente descripción, comprendiendo el decodificador una unidad (3001) de obtención configurada para obtener un valor de un índice de línea de referencia de intrapredicción de un bloque de transformación; y una unidad (3003) de predicción configurada para obtener muestras de predicción del bloque de transformación basándose en muestras de referencia de una línea de referencia, en donde la línea de referencia está indexada por el valor del índice de línea de referencia de intrapredicción.

El decodificador 30 puede incluir una unidad 3002 de selección (no mostrada en la FIG. 16), configurada para seleccionar la línea de muestras de referencia basándose en el valor del índice de línea de referencia de intrapredicción, como los ejemplos mostrados en las FIGs. 9-13.

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. 17 es un diagrama de bloques que muestra un sistema 3100 de suministro de contenido para realizar el servicio de distribución de contenido. Este sistema 3100 de suministro de contenidos incluye un dispositivo 3102 de captura, un dispositivo 3106 terminal y, opcionalmente, incluye una pantalla 3126. El dispositivo 3102 de captura se comunica con el dispositivo 3106 terminal a través del enlace 3104 de comunicación. El enlace de comunicación puede incluir el canal 13 de comunicación descrito anteriormente. El enlace 3104 de comunicación incluye, pero no se limita a, WIFI, Ethernet, cable, inalámbrico (3G/4G/5G), USB o cualquier tipo de combinación de los mismos, o similares.

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

En el sistema 3100 de suministro de contenidos, el dispositivo 310 terminal recibe y reproduce los datos codificados. El dispositivo 3106 terminal podría ser un dispositivo con capacidad de recepción y recuperación de datos, tal como un teléfono inteligente o Pad 3108, un ordenador u ordenador portátil 3110, una grabadora de vídeo en red (NVR)/grabadora de vídeo digital (DVR) 3112, TV 3114, decodificador (STB) 3116, sistema 3118 de videoconferencia, sistema 3120 de videovigilancia, asistente digital personal (PDA) 3122, dispositivo 3124 montado en vehículo, o una combinación de cualquiera de ellos, o similar capaz de decodificar los datos codificados mencionados anteriormente. Por ejemplo, el dispositivo 3106 terminal puede incluir el dispositivo 14 de destino como se describió anteriormente. Cuando los datos codificados incluyen vídeo, se da prioridad al decodificador 30 de vídeo incluido en el dispositivo terminal para realizar la decodificación de vídeo. Cuando los datos codificados incluyen audio, se da prioridad a 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 Pad 3108, ordenador u ordenador portátil 3110, grabadora de vídeo en red (NVR)/grabadora de vídeo digital (DVR) 3112, TV 3114, asistente digital personal (PDA) 3122, o dispositivo 3124 montado en vehículo, el dispositivo terminal puede enviar los datos decodificados a su pantalla. Para un dispositivo terminal equipado sin pantalla, por ejemplo, STB 3116, sistema 3118 de videoconferencia o sistema 3120 de videovigilancia, se contacta con una pantalla 3126 externa para recibir y mostrar los datos decodificados.

Cuando cada dispositivo en este sistema realiza 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. 18 es un diagrama que muestra una estructura de un ejemplo del dispositivo 3106 terminal. Después

de que el dispositivo 3106 terminal recibe el flujo del dispositivo 3102 de captura, la unidad 3202 de procedimiento de protocolo analiza el protocolo de transmisión del flujo. El protocolo incluye, pero no se limita, al protocolo de transmisión en tiempo real (RTSP), al protocolo de transferencia de hipertexto (HTTP), al protocolo de transmisión en vivo HTTP (HLS), a MPEG-DASH, al protocolo de transporte en tiempo real (RTP), al protocolo de mensajería en tiempo real (RTMP), o a cualquier tipo de combinación de los mismos, o similares.

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

A través del procesamiento de demultiplexación, se generan flujos elementales de vídeo (ES), ES de audio y, opcionalmente, subtítulos. El decodificador 3206 de vídeo, que incluye el decodificador 30 de vídeo como se explica en las realizaciones mencionadas anteriormente, decodifica el vídeo ES mediante el método de decodificación como se muestra en las realizaciones mencionadas anteriormente para generar tramas de vídeo y envía estos datos a la unidad 3212 síncrona. El decodificador 3208 de audio decodifica el ES de audio para generar una trama de audio y envía estos datos a la unidad 3212 síncrona. Alternativamente, la trama de vídeo puede almacenarse en un búfer (no mostrado en la FIG. Y) antes de enviarlo a la unidad 3212 síncrona. De manera similar, la trama de audio puede almacenarse en un búfer (no mostrado en la FIG. Y) antes de enviarlo a la unidad 3212 síncrona.

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

Si se incluyen subtítulos en el flujo, el decodificador 3210 de subtítulos decodifica el subtítulo y lo sincroniza con la trama de vídeo y la trama de audio, y suministra el vídeo/audio/subtítulo a una pantalla 3216 de vídeo/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 a otro sistema, por ejemplo, un sistema de automóvil.

Operadores Matemáticos

Los operadores matemáticos usados en esta solicitud son similares a los usados en el lenguaje de programación C. Sin embargo, los resultados de la división de números enteros y las operaciones de desplazamiento aritmético se definen con mayor precisión, y se definen operaciones adicionales, tal como la exponenciación y la división de 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:

- + 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 la potencia de y. En otros contextos, dicha notación se usa para superíndices que no están destinados a ser interpretados como exponenciación.
- / División de números 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.
- $\div$ Se usa para indicar división en ecuaciones matemáticas donde no se pretende truncar ni redondear.
- $\frac{x}{y}$ Se usa para indicar división en ecuaciones matemáticas donde no se pretende truncar ni redondear.

$\sum_{i=x}^y f(i)$ El sumatorio de $f(i)$ con i tomando todos los valores de números enteros desde x hasta y inclusive.

$x\%y$ Módulo. Resto de x dividido por y , definido solo para números enteros x e y con $x \geq 0$ e $y > 0$.

Operadores lógicos

Los siguientes operadores lógicos se definen de la siguiente manera:

$x \&\& y$ Lógico booleano "and" de x e y

$x || y$ Lógico booleano "or" de x e y

! Lógico booleano "not"

$x ? y : z$ Si x es VERDADERO o no es igual a 0, se evalúa como el valor de y ; de lo contrario, se evalúa como el valor de z .

5

Operadores relacionales

Los siguientes operadores relacionales se definen de la siguiente manera:

> Mayor que

>= Mayor o igual que

< Menor que

<= Menor o igual que

= = Igual a

!= No igual a

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

Operadores bit a bit

15

Los siguientes operadores bit a bit se definen de la siguiente manera:

&	"and" bit a bit. Cuando opera con argumentos de números enteros, opera con una representación en complemento a dos del valor de número entero. Cuando opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se extiende añadiendo bits más significativos iguales a 0.
	"or" bit a bit. Cuando opera con argumentos de números enteros, opera con una representación en complemento a dos del valor de número entero. Cuando opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se extiende añadiendo bits más significativos iguales a 0.
^	"or exclusivo" bit a bit. Cuando opera con argumentos de números enteros, opera con una representación en complemento a dos del valor de número entero. Cuando opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se extiende añadiendo bits más 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 se define solo para valores de números enteros no negativos de y. Los bits desplazados a los bits más significativos (MSB) 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 se define solo para valores de números enteros no negativos de y. Los bits desplazados a los bits menos significativos (LSB) 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, se evalúa como 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, se evalúa como el valor de la variable antes de la operación de decremento.
- += Incrementar por la cantidad especificada, es decir, $x += 3$ es equivalente a $x = x + 3$, y $x += (-3)$ es equivalente a $x = x + (-3)$.
- = Decrementar en la cantidad especificada, es decir, $x -= 3$ equivale a $x = x - 3$, y $x -= (-3)$ equivale a $x = x - (-3)$.

5

Notación de intervalo

La siguiente notación se usa para especificar un intervalo de valores:

- 10 $x = y..z$ x toma valores de números enteros comenzando desde y hasta z , inclusive, siendo x , y , y z números enteros y siendo z mayor que y .

Funciones matemáticas

- 15 Se definen las siguientes funciones matemáticas:

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

- 20 $\text{Asin}(x)$ la función trigonométrica de seno inverso, que opera con un argumento x que está en el intervalo de $-1,0$ a $1,0$, inclusive, con un valor de salida en el intervalo de $-\pi/2$ a $\pi/2$, inclusive, en unidades de radianes

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

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

25

$\text{Ceil}(x)$ el número entero más pequeño mayor o igual que x .

$$\text{Clip1}_V(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_V) - 1, x)$$

30

$$\text{Clip1}_C(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_C) - 1, x)$$

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

- 35 $\text{Cos}(x)$ la función trigonométrica de coseno que opera sobre un argumento x en unidades de radianes.

$\text{Floor}(x)$ el número entero más grande menor o igual que x .

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

40

$\ln(x)$ el logaritmo natural de x (el logaritmo en base e , donde e es la constante de base del logaritmo natural 2,718 281 828...).

$\log_2(x)$ el logaritmo de base 2 de x .

5

$\log_{10}(x)$ el logaritmo en base 10 de x .

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

10

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

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0.5)$$

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

15

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

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

20

$$\text{Swap}(x, y) = (y, x)$$

$\tan(x)$ la función trigonométrica de tangente que opera sobre un argumento x en unidades de radianes.

Orden de precedencia de operación

25

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

30

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

35

Para aquellos operadores que también se usan en el lenguaje de programación C, el orden de precedencia usado en esta Memoria Descriptiva es el mismo que se usa en el lenguaje de programación C.

40

Tabla: Prioridad de operaciones desde la más alta (en la parte superior de la tabla) hasta la más baja (en la parte inferior de la tabla)

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

"x && y"
"x y"
"x ? y : z"
"x..y"
"x = y ", "x += y", "x -= y"

Descripción textual de operaciones lógicas.

5 En el texto, una declaración de operaciones lógicas como se describiría matemáticamente de la siguiente forma:

Si (condición 0)

declaración 0

10

de lo contrario si (condición 1)

declaración 1

15

...

de lo contrario /* comentario informativo sobre la condición restante */

declaración n

20

se puede describir de la siguiente manera:

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

25

– Si condición 0, declaración 0

– De lo contrario, si la condición 1, declaración 1

– ...

30

– En caso contrario (observación informativa sobre la condición restante), declaración n

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

35

En el texto, una declaración de operaciones lógicas como se describiría matemáticamente de la siguiente forma:

40

Si (condición 0a && condición 0b)

declaración 0

45

de lo contrario si (condición 1a || condición 1b)

declaración 1

50

...

de lo contrario

declaración n

55

se puede describir de la siguiente manera:

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

- 5 – Si todas las condiciones siguientes son verdaderas, declaración 0:
- Condición 0a
- Condición 0b
- 10 – En caso contrario, si una o más de las siguientes condiciones son verdaderas, declaración 1:
- condición 1a
- condición 1b
- 15 – ...
- De lo contrario, declaración n
- 20 En el texto, una declaración de operaciones lógicas como se describiría matemáticamente de la siguiente forma:
- si (condición 0)
- 25 declaración 0
- si (condición 1)
- declaración 1
- 30 se puede describir de la siguiente manera:
- Cuando condición 0, declaración 0
- 35 Cuando condición 1, declaración 1
- 40 Aunque las realizaciones de la invención se han descrito principalmente basándose en la codificación de vídeo, cabe señalar que las realizaciones del sistema 10 de codificación, el codificador 20 y el decodificador 30 (y correspondientemente el sistema 10) y las otras realizaciones descritas en la presente memoria también pueden configurarse para procesamiento o codificación de imágenes fijas, es decir, el procesamiento o codificación de una imagen individual independiente de cualquier imagen anterior o consecutiva como en la codificación de vídeo. En general, solo las unidades 244 de interpretación (codificador) y 344 (decodificador) pueden no estar disponibles en caso de que la codificación de procesamiento de imágenes se limite a una única imagen 17. Todas las demás funcionalidades (también denominadas herramientas o tecnologías) del
- 45 codificador 20 de vídeo y del decodificador 30 de vídeo pueden usarse igualmente para el procesamiento de imágenes fijas, por ejemplo, cálculo 204/304 residual, transformación 206, cuantificación 208, cuantificación 210/310 inversa, transformación 212/312 (inversa), partición 262/362, intrapredicción 254/354 y/o filtrado 220, 320 de bucle y codificación 270 de entropía y decodificación 304 de entropía.
- 50 Las realizaciones, por ejemplo, del codificador 20 y el decodificador 30, y las funciones descritas en la presente memoria, por ejemplo, con referencia al codificador 20 y el decodificador 30, pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Cuando se implementan en software, estas funciones pueden almacenarse en un soporte legible por ordenador o transmitido sobre soportes de comunicación como una o más instrucciones o códigos ejecutados mediante una unidad de procesamiento basada en hardware. Los soportes legibles por ordenador pueden incluir soportes de almacenamiento legibles por ordenador, que corresponden a un soporte tangible tal como soportes de almacenamiento de datos, o soportes de comunicación que incluyen cualquier soporte que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, según un protocolo de comunicación. De esta manera, los soportes legibles por ordenador generalmente pueden corresponder a (1) soportes de almacenamiento tangibles legibles por ordenador que no son transitorios o (2) un soporte de comunicación tal como una señal u onda portadora.
- 60 Los soportes de almacenamiento de datos pueden ser cualquier soporte disponible al que puedan acceder uno o más ordenadores o uno o más procesadores para recuperar instrucciones, códigos y/o estructuras de datos para la implementación de las técnicas descritas en esta descripción. Un producto de programa informático puede incluir un soporte legible por ordenador.

- A modo de ejemplo, y sin limitación, tales soportes 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 de semiconductores, o cualquier otro soporte que pueda usarse para almacenar el código de programa deseado en la forma de instrucciones o estructuras de datos y al que pueda acceder un ordenador. Además, cualquier conexión se denomina propiamente un soporte legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un cable de par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, DSL o tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de soporte. Sin embargo, debe entenderse que los soportes de almacenamiento legibles por computadora y los soportes de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros soportes transitorios, sino que están dirigidos a soportes de almacenamiento tangibles y no transitorios. El disco y disquete usados en la presente memoria incluyen un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disquete y un disco blu-ray, donde los discos generalmente reproducen datos magnéticamente, mientras que los disquetes reproducen datos ópticamente mediante láser. Las combinaciones de lo anterior también deben incluirse en el alcance de los soportes legibles por ordenador.
- Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), matrices lógicas programables en campo (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", tal como se usa en la presente memoria, puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en la presente memoria. Además, en algunos aspectos, la funcionalidad descrita en la presente memoria puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o incorporados en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.
- Las técnicas de esta descripción se pueden implementar en una amplia variedad de dispositivos o aparatos, incluido un teléfono inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). En esta descripción se describen diversos componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas descritas, pero no necesariamente requieren la realización mediante diferentes unidades de hardware. Más bien, como se describió anteriormente, se pueden combinar diversas unidades en una unidad de hardware de códec o proporcionarse mediante una colección de unidades de hardware interoperativas, incluyendo uno o más procesadores como se describió anteriormente, junto con software y/o firmware adecuado.

REIVINDICACIONES

1. Un método de intrapredicción del modo de intrapredicción DC para un bloque, que comprende:

5 obtener (1601) un valor de un índice de línea de referencia de intrapredicción, refIdx, de un bloque de transformación; y

obtener (1603) muestras de predicción del bloque de transformación basándose en muestras de referencia p[x] [y] de una línea de referencia, en donde la línea de referencia está indexada por el valor del índice de línea de referencia de intrapredicción refIdx, siendo el valor de refIdx distinto de cero;

10 en donde las muestras de predicción del bloque de transformación predSamples[x] [y] se obtienen mediante:

predSamples[x] [y] = dcVal, con x = 0.. nTbW-1, y = 0.. nTbH-1

15

en donde cuando nTbW es igual a nTbH, la variable dcVal se obtiene mediante:

$$dcVal = (\sum_{x'=0}^{nTbW-1} p[x'] [-1 - refIdx] + \sum_{y'=0}^{nTbH-1} p[-1 - refIdx][y'] + nTbW) \gg (\text{Log2}(nTbW) + 1);$$

20

en donde cuando el ancho del bloque de transformación es mayor que la altura del bloque de transformación, solo se usan muestras de referencia de una línea de referencia superior para obtener las muestras de predicción, y la variable dcVal se obtiene mediante:

$$dcVal = (\sum_{x'=0}^{nTbW-1} p[x'] [-1 - refIdx] + (nTbW \gg 1)) \gg \text{Log2}(nTbW);$$

25

en donde cuando la altura del bloque de transformación es mayor que la anchura del bloque de transformación, solo se usan muestras de referencia de una línea de referencia izquierda para obtener las muestras de predicción, y la variable dcVal se obtiene mediante:

30

$$dcVal = (\sum_{y'=0}^{nTbH-1} p[-1 - refIdx][y'] + (nTbH \gg 1)) \gg \text{Log2}(nTbH);$$

p[x'] [-1 - refIdx] que especifica las muestras de referencia desde una línea de referencia superior,

35 p[-1 - refIdx] [y'] que especifica las muestras de referencia desde una línea de referencia izquierda,

una variable nTbW que especifica un ancho del bloque de transformación,

una variable nTbH que especifica una altura del bloque de transformación;

40

y

en donde las muestras de referencia se determinan especificando un índice horizontal o vertical, cada uno de los índices es un valor de número entero que pertenece a un intervalo de valores de números enteros, y en donde los índices especifican una coordenada horizontal (x') o una coordenada vertical (y') dentro la línea de referencia, respectivamente; y

45

en donde la muestra de referencia más superior de la línea de referencia se desplaza verticalmente a la posición de la muestra predicha superior izquierda del bloque de transformación, y en donde el valor del desplazamiento es igual al valor del índice de línea de referencia de intrapredicción; y

50

la muestra de referencia más izquierda de la línea de referencia se desplaza horizontalmente a la posición de la muestra predicha superior izquierda del bloque de transformación, y en donde el valor del desplazamiento es igual al valor del índice de línea de referencia de intrapredicción.

55

2. El método de la reivindicación 1, en donde el valor del índice de línea de referencia de intrapredicción, refIdx, se indica mediante

intra_luma_ref_idx.

60

3. El método de cualquiera de las reivindicaciones 1-2, en donde el valor del índice de línea de referencia de intrapredicción es 1 o 2.

4. El método de una cualquiera de las reivindicaciones 1-3, en donde las muestras de referencia forman un intervalo contiguo o una distancia contigua dentro de la línea de referencia.

5 5. Un decodificador (30) o un codificador (20), que comprende:

uno o más procesadores; y

10 un soporte de almacenamiento legible por ordenador no transitorio acoplado a los procesadores y que almacena la programación para su ejecución por los procesadores, en donde la programación, cuando la ejecutan los procesadores, configura el decodificador o el codificador para llevar a cabo el método según una cualquiera de las reivindicaciones 1 a 4.

15 6. Un soporte legible por ordenador no transitorio que transporta un código de programa que, cuando es ejecutado por un dispositivo informático, hace que el dispositivo informático realice el método de una cualquiera de las reivindicaciones 1 a 4.

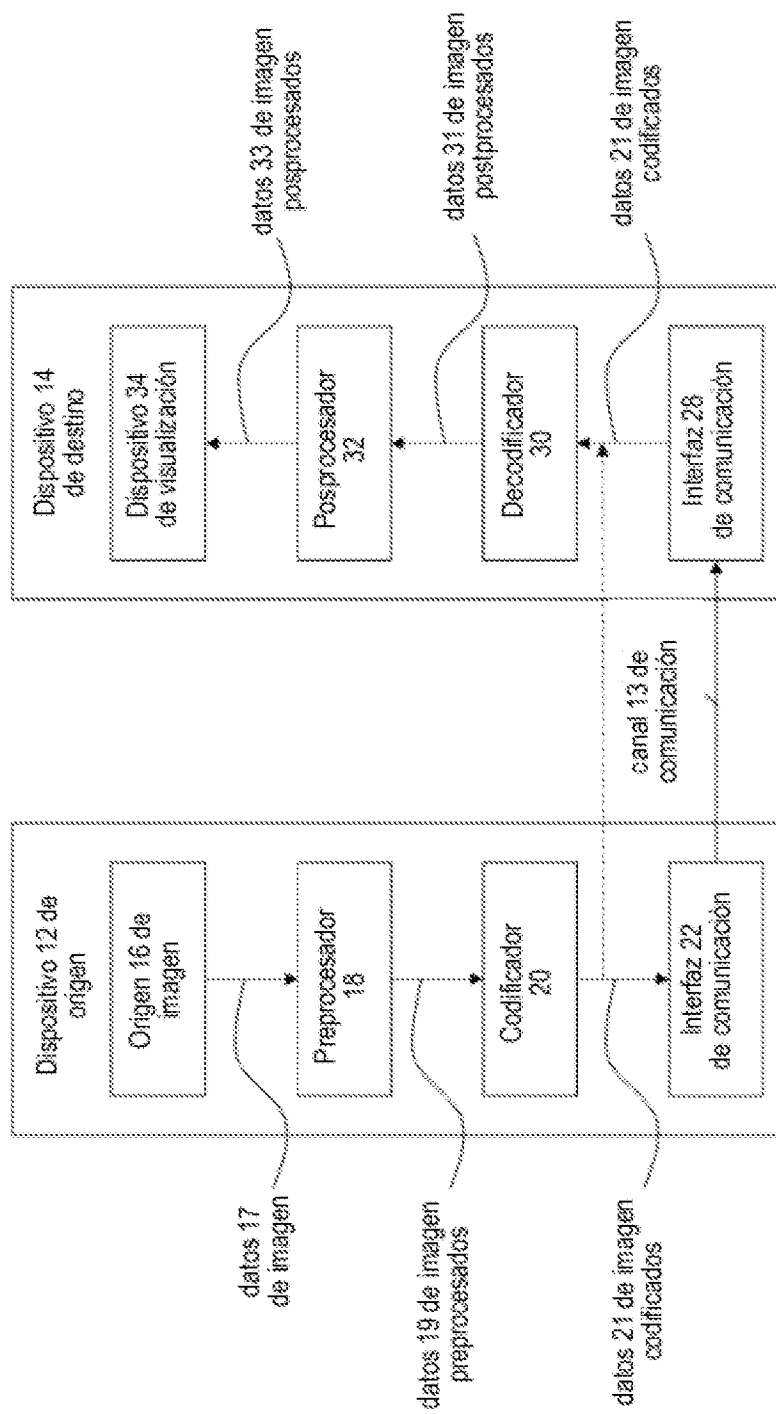


Fig. 1A

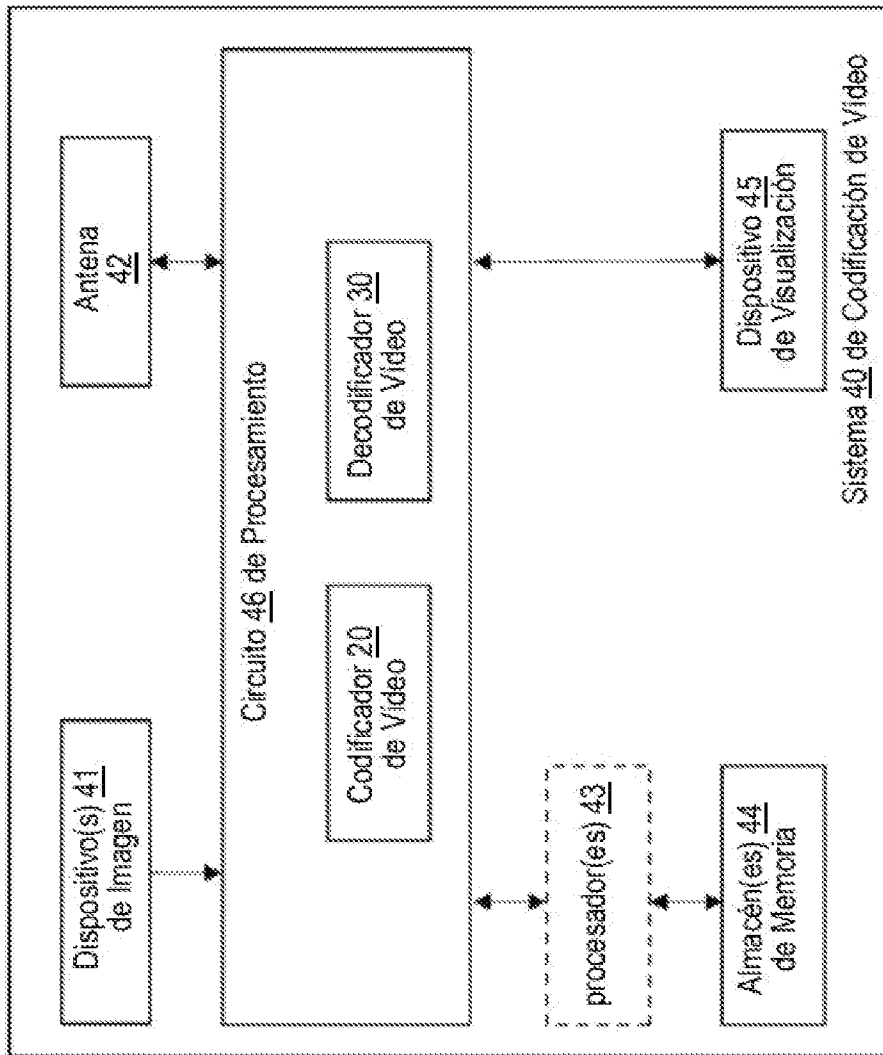


Fig. 1B

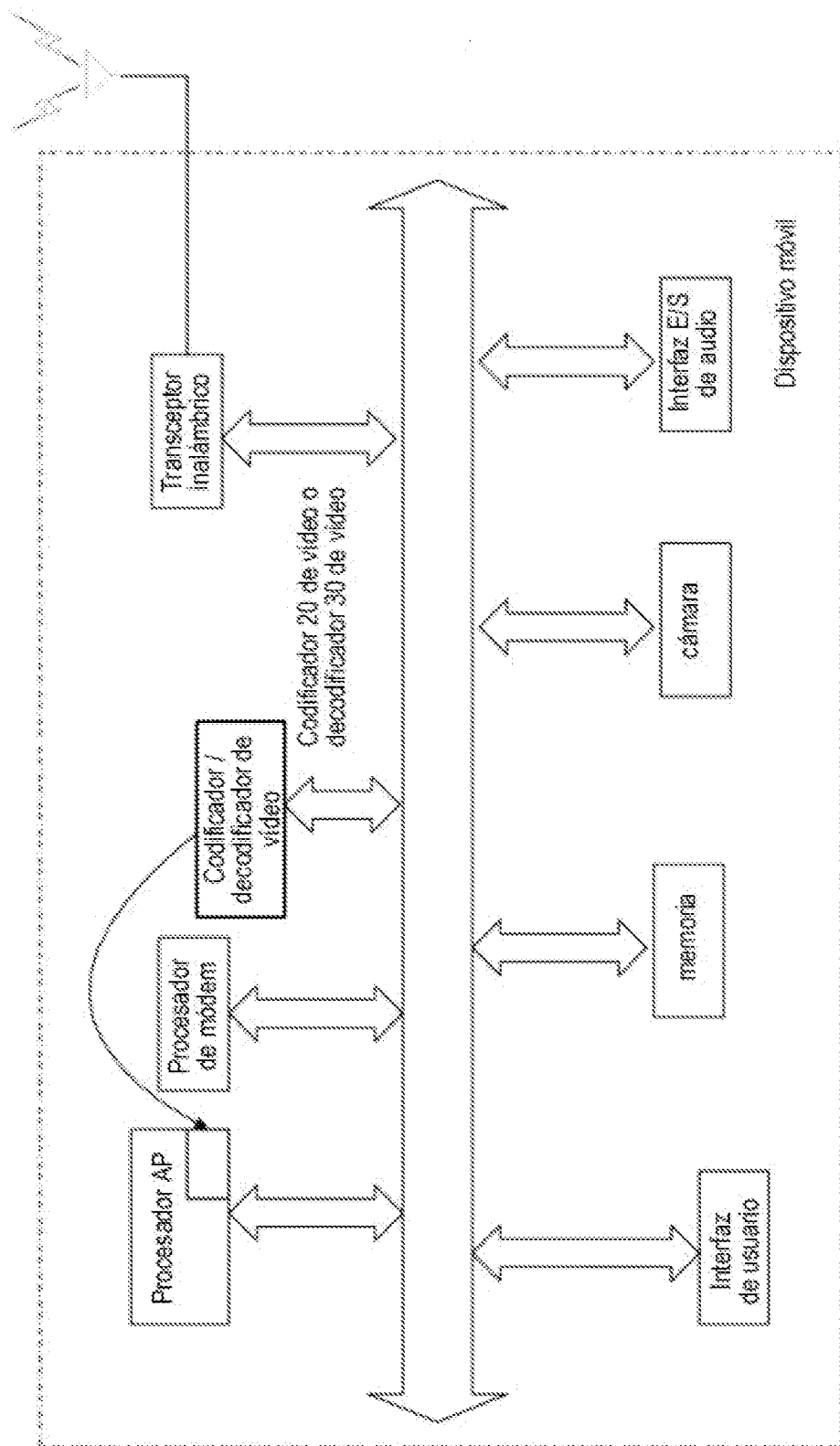
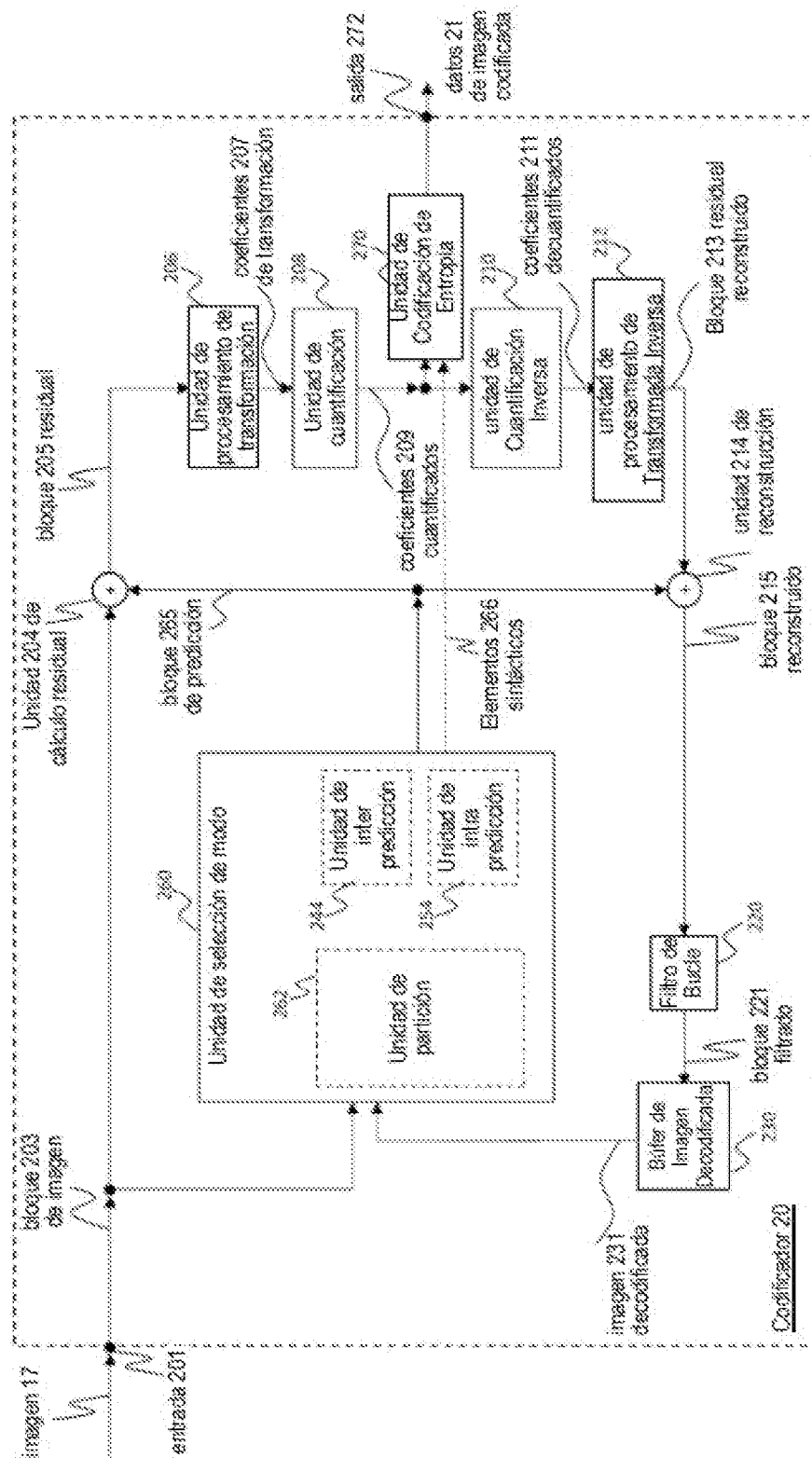


Fig. 1C



236

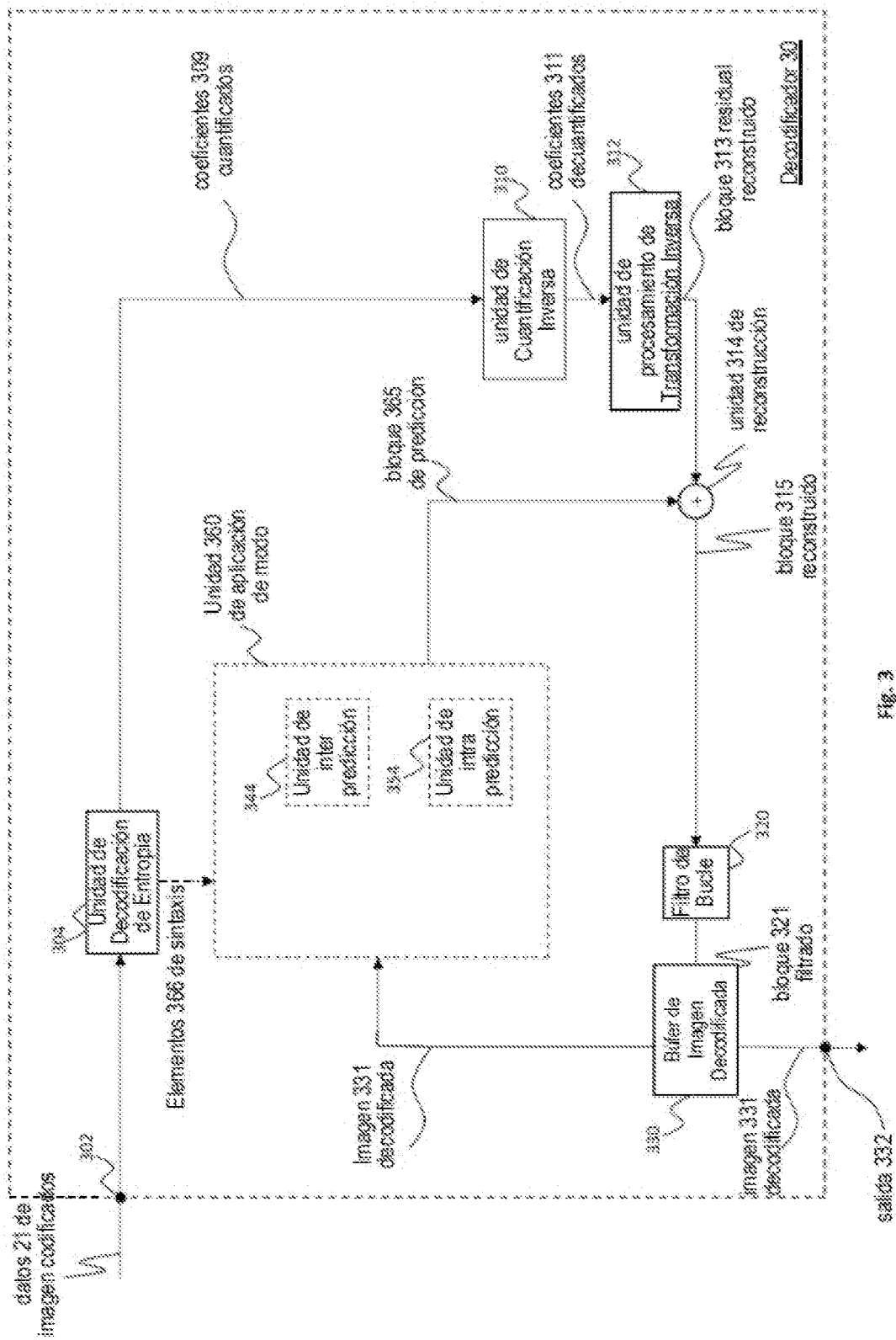


Fig. 3

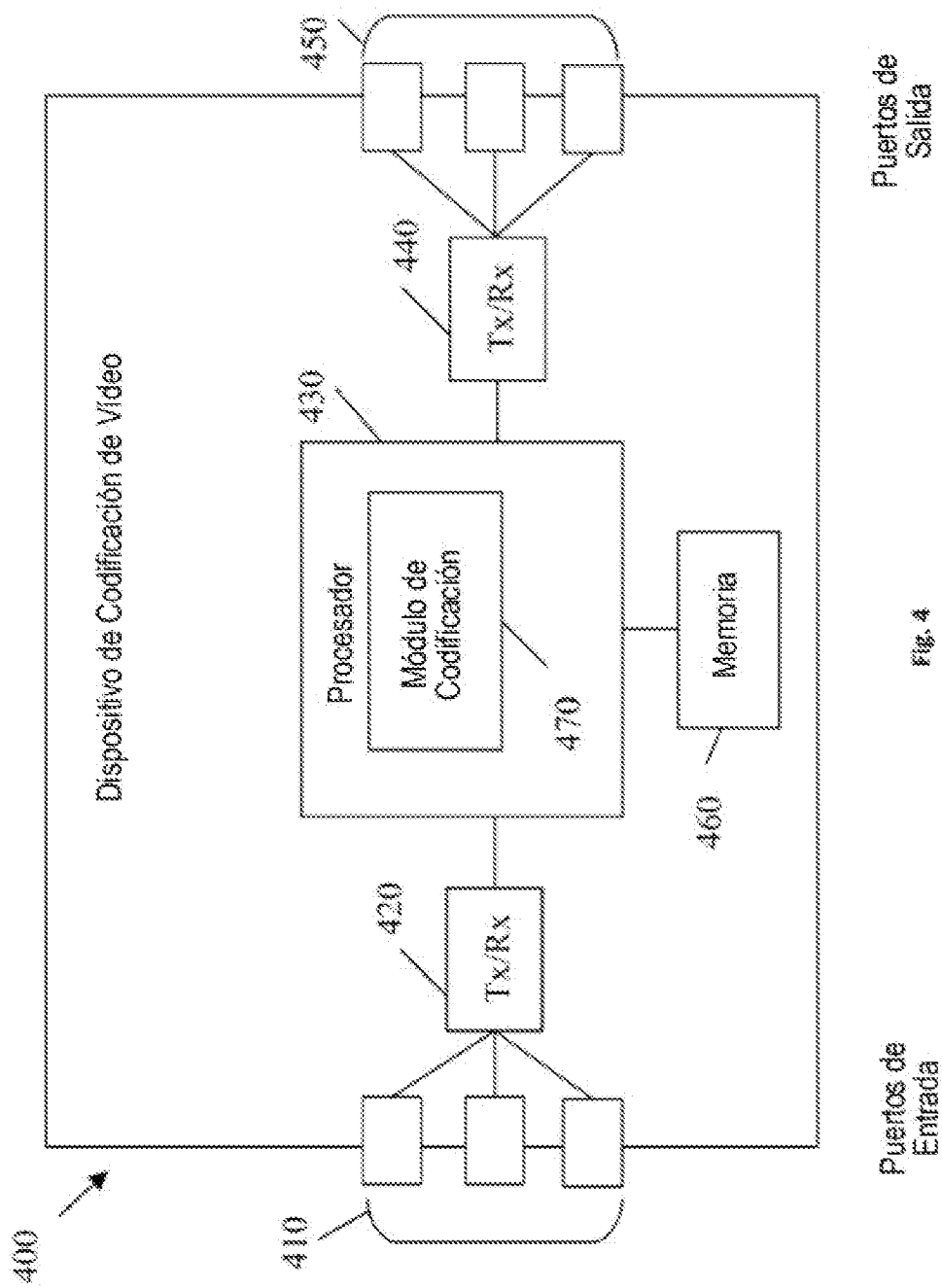


Fig. 4

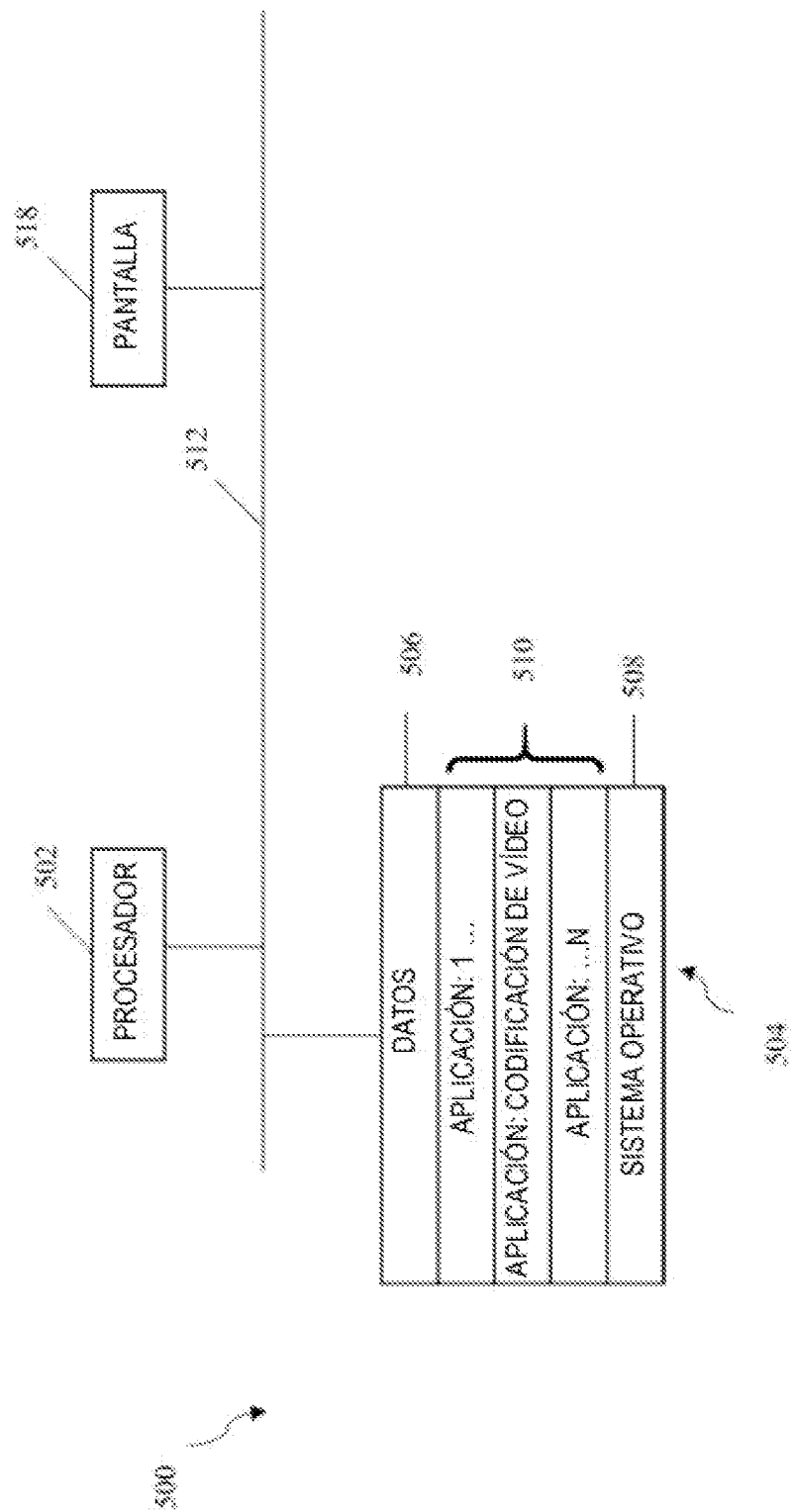


Fig. 5

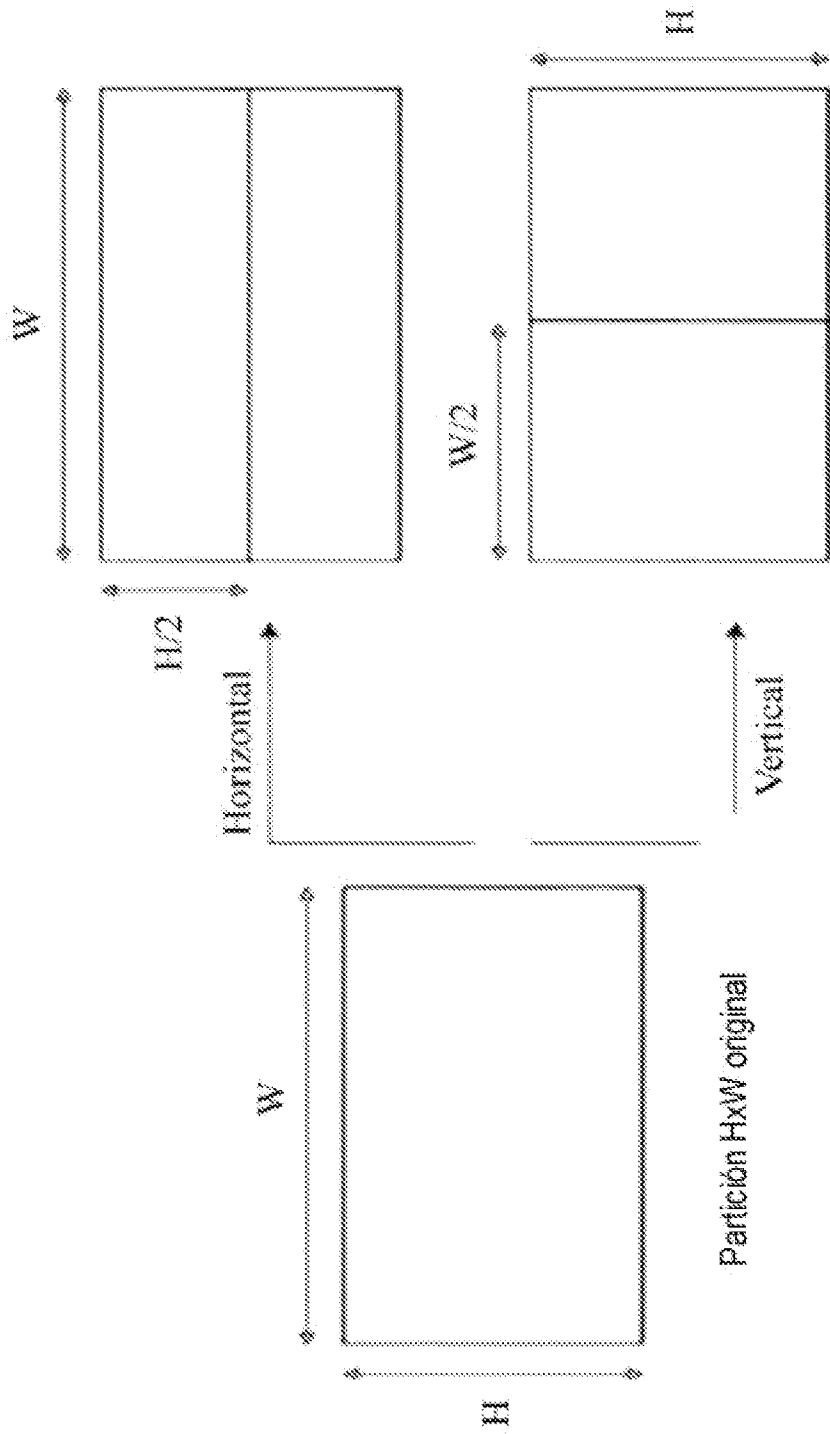


FIG. 6

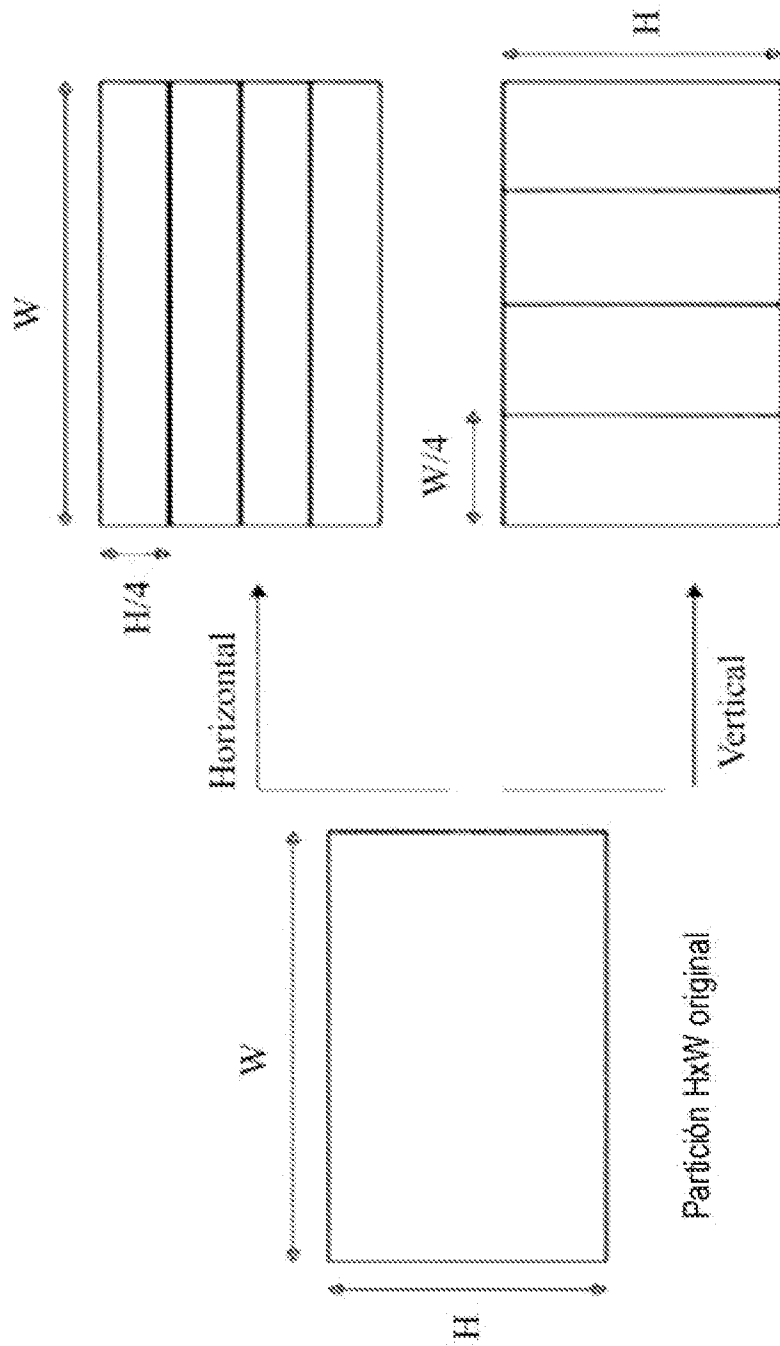
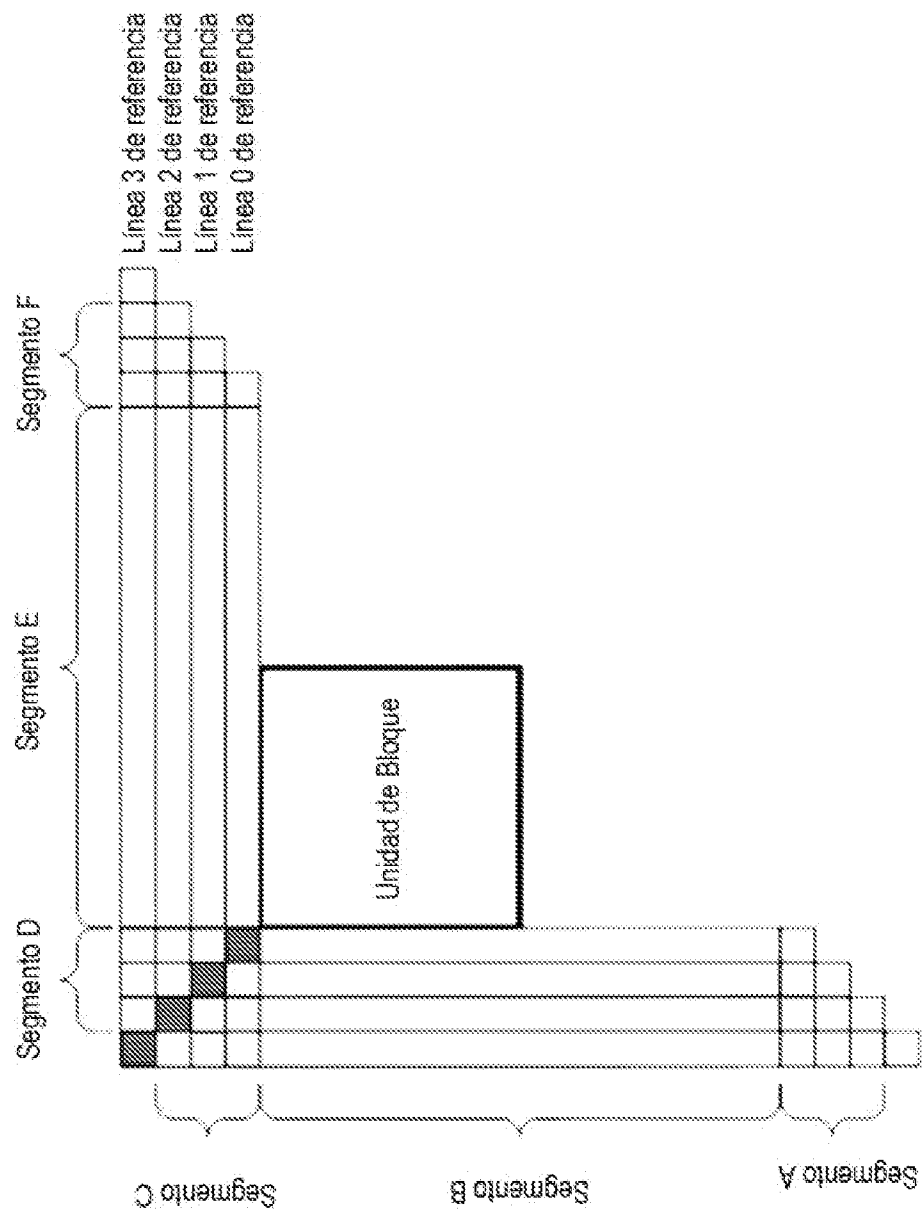


FIG. 7



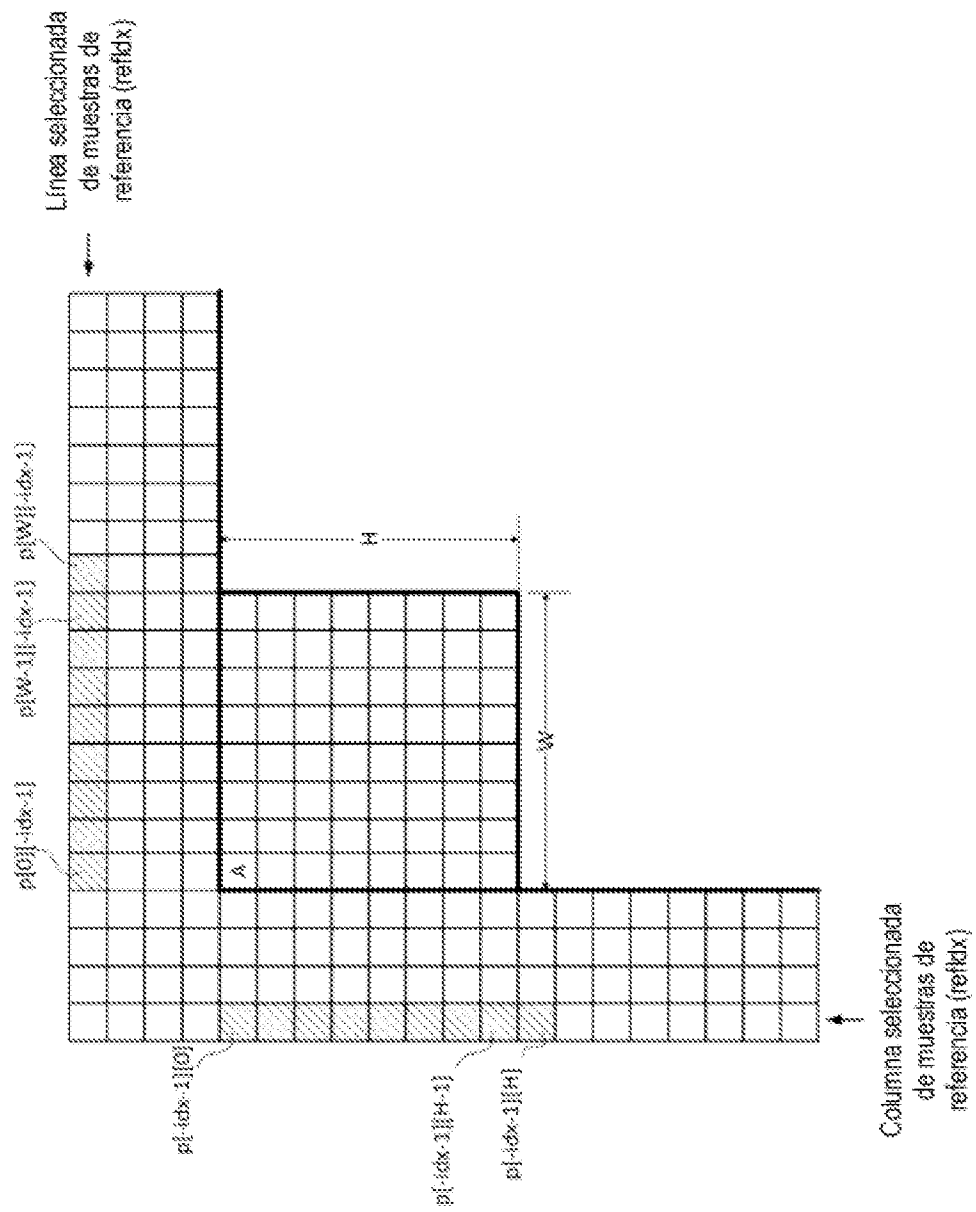


FIG. 9

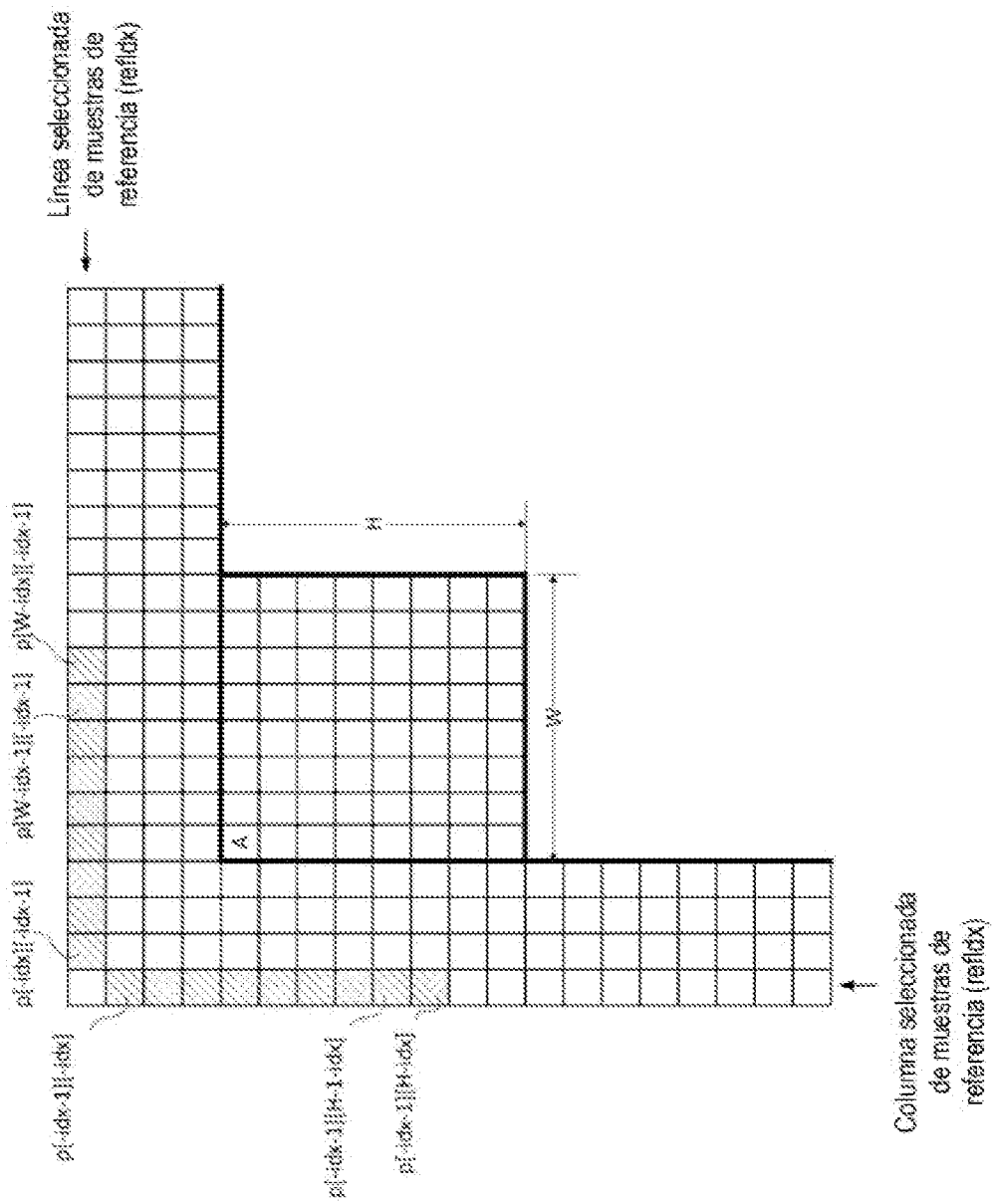


FIG. 10

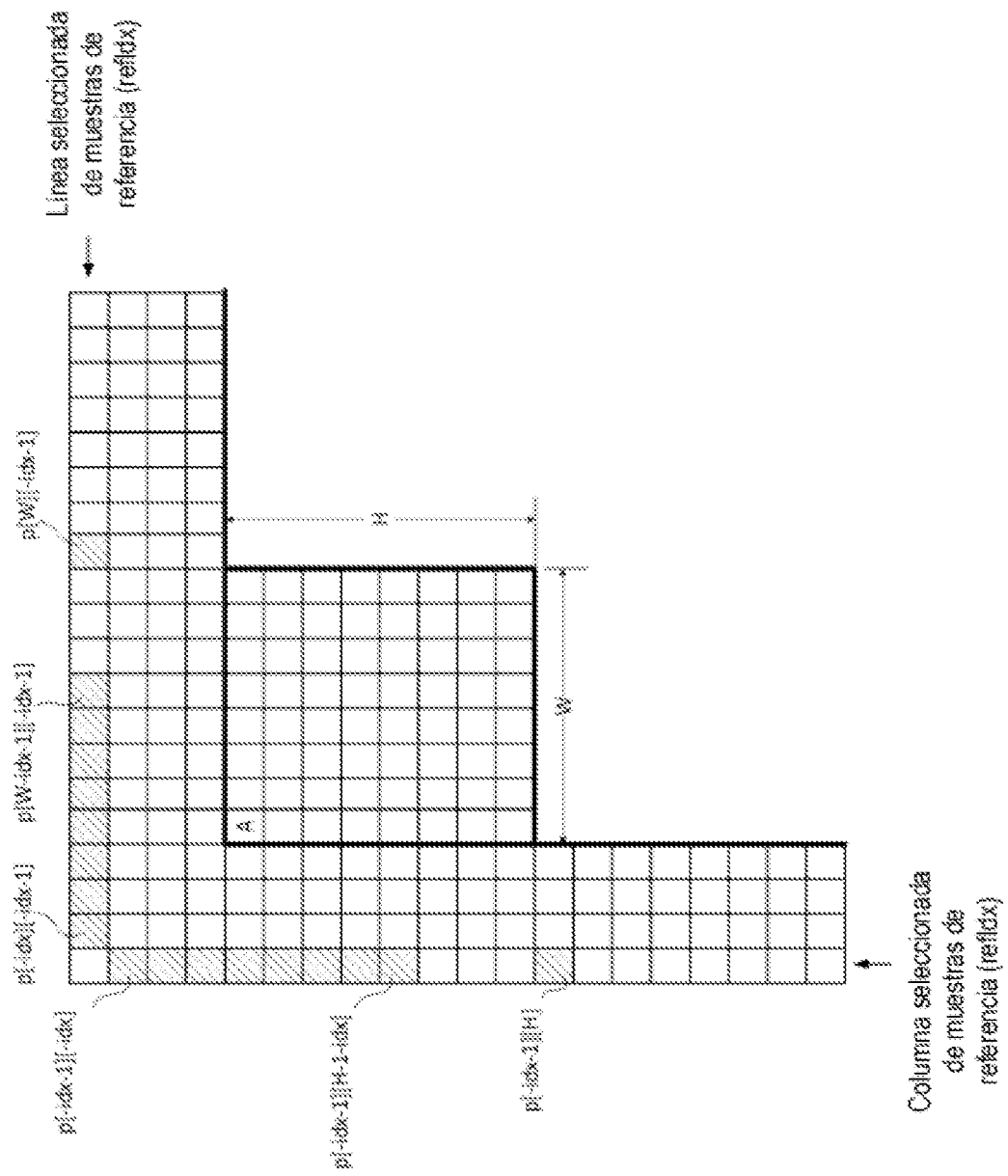


FIG. 11

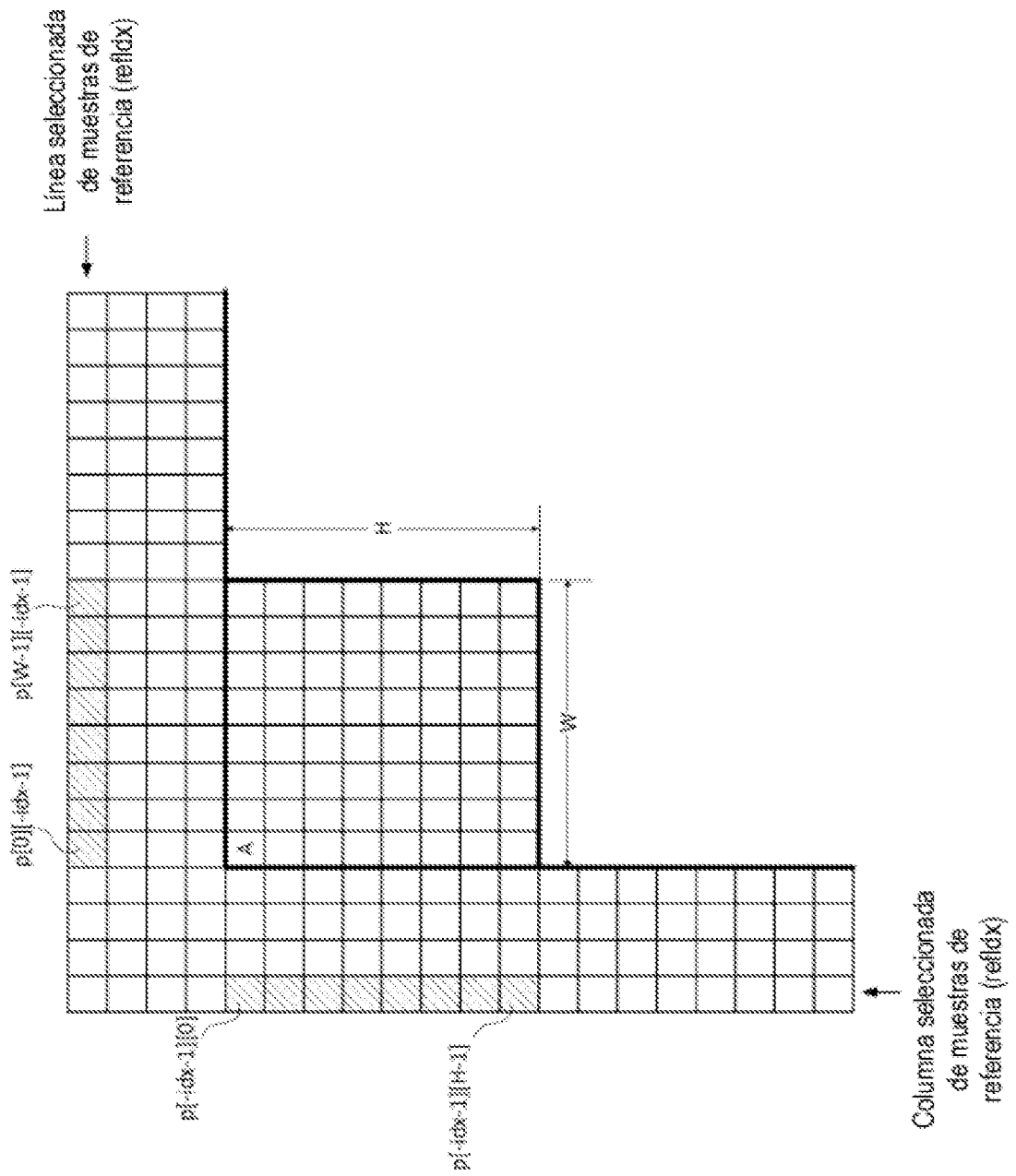


FIG. 12

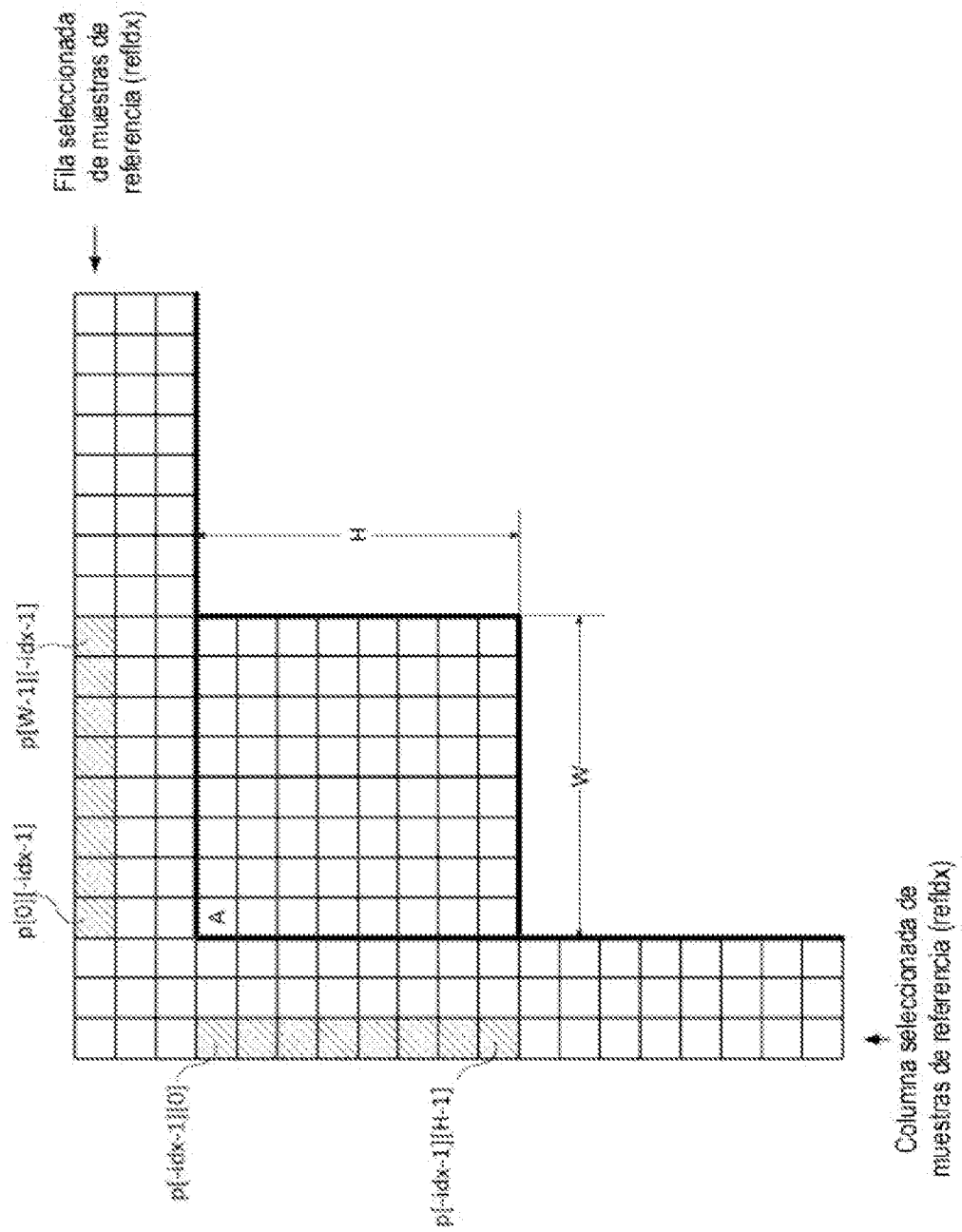


FIG. 12A

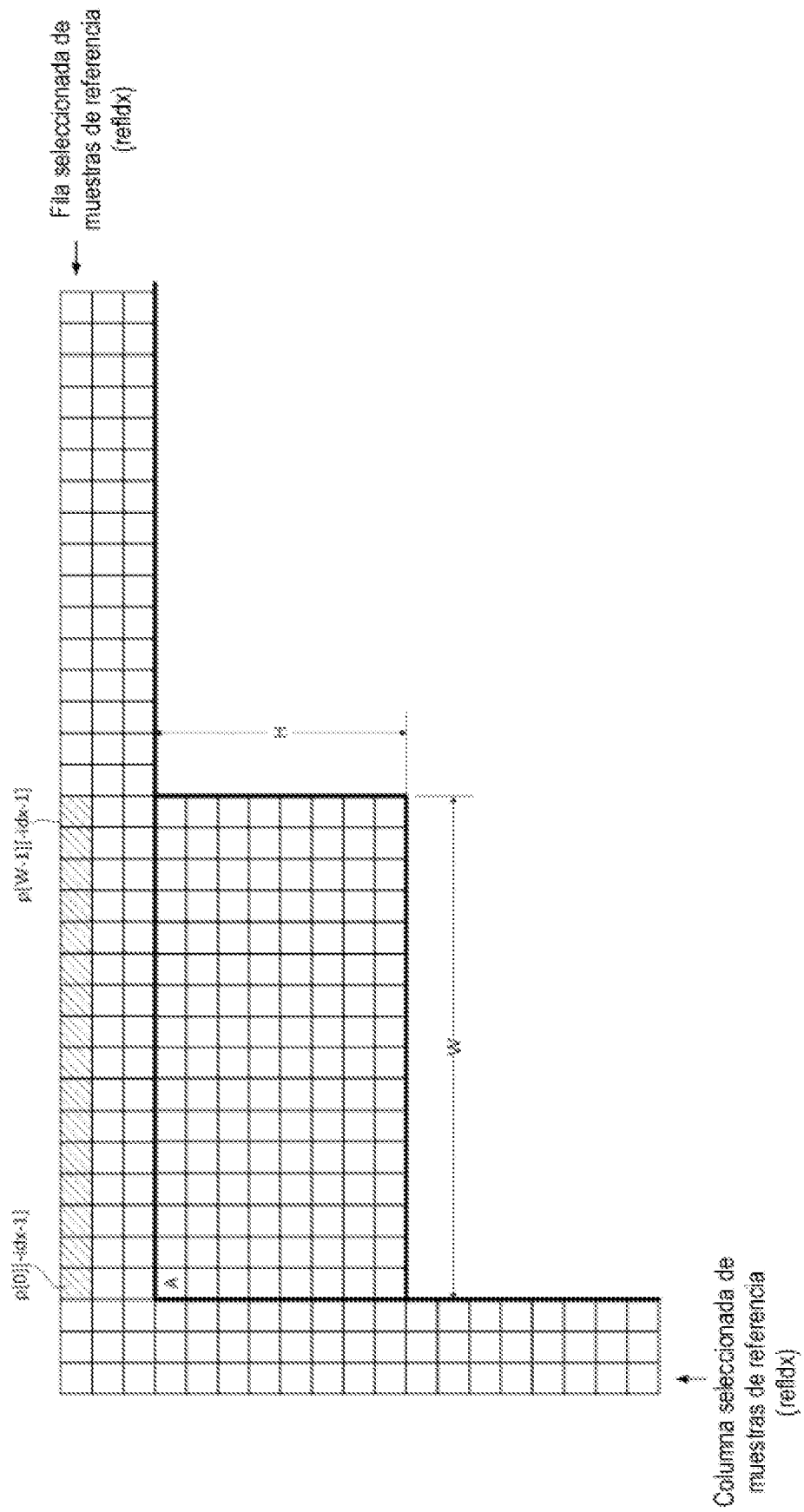


FIG. 12B

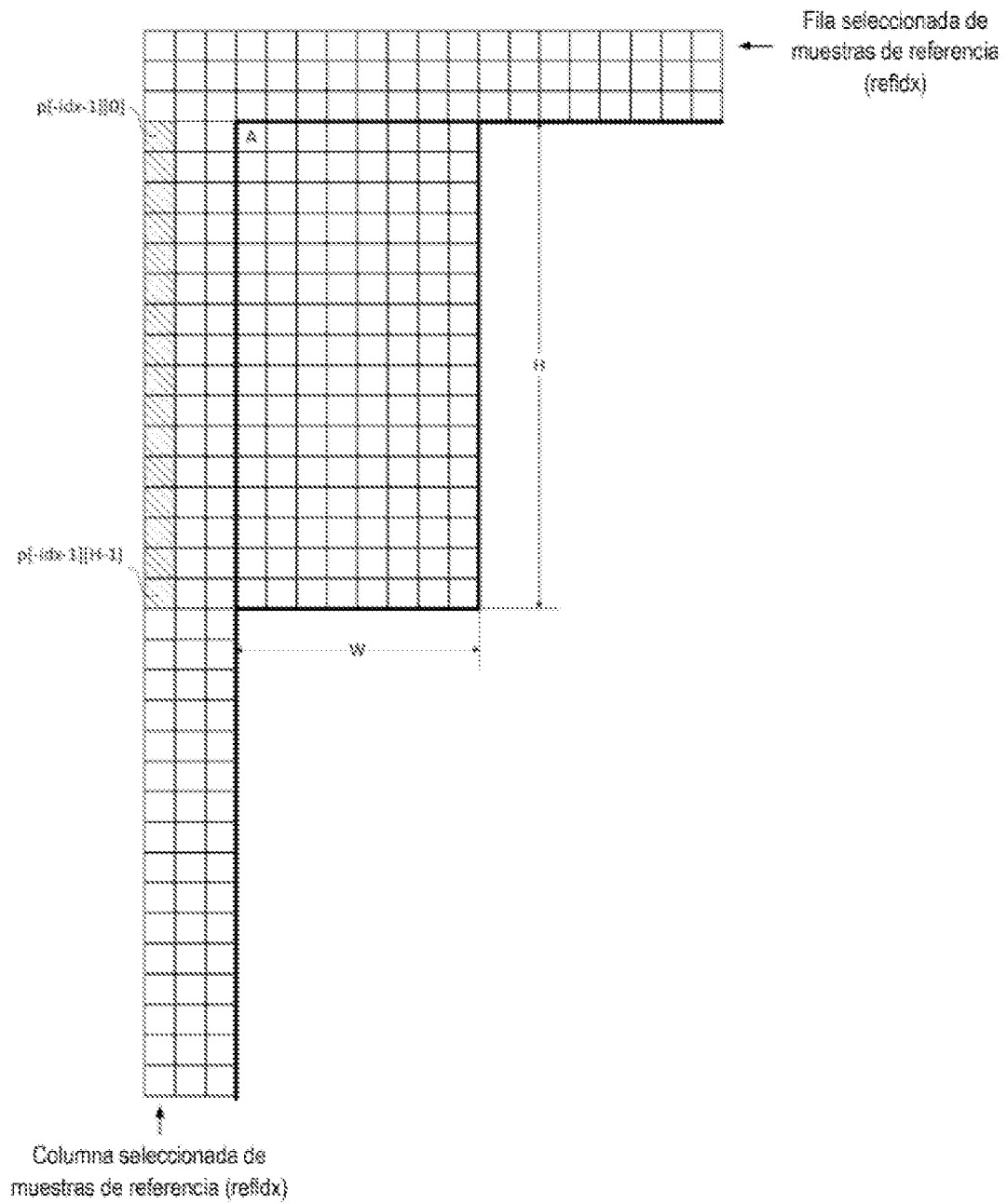
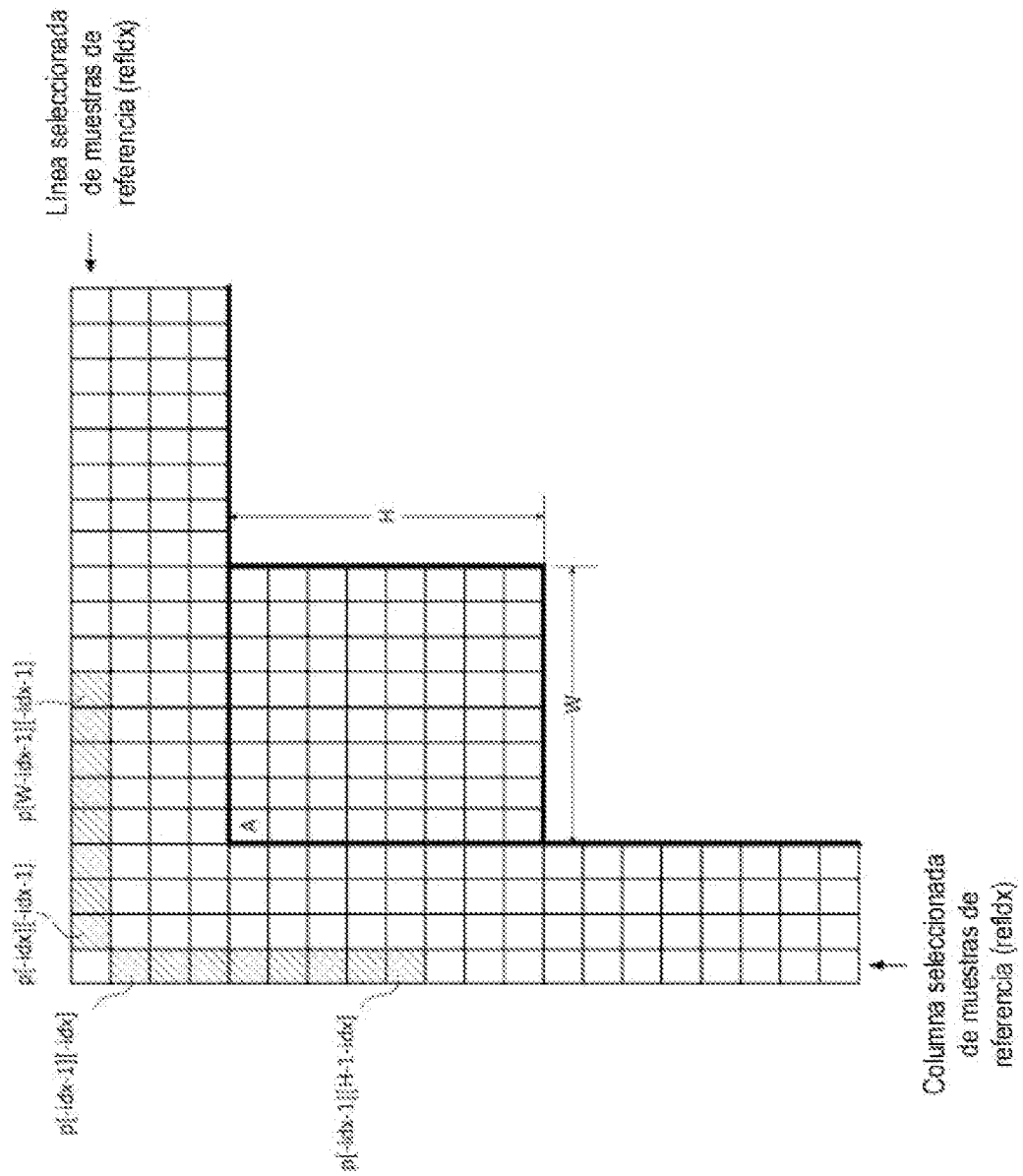


FIG. 12C



30

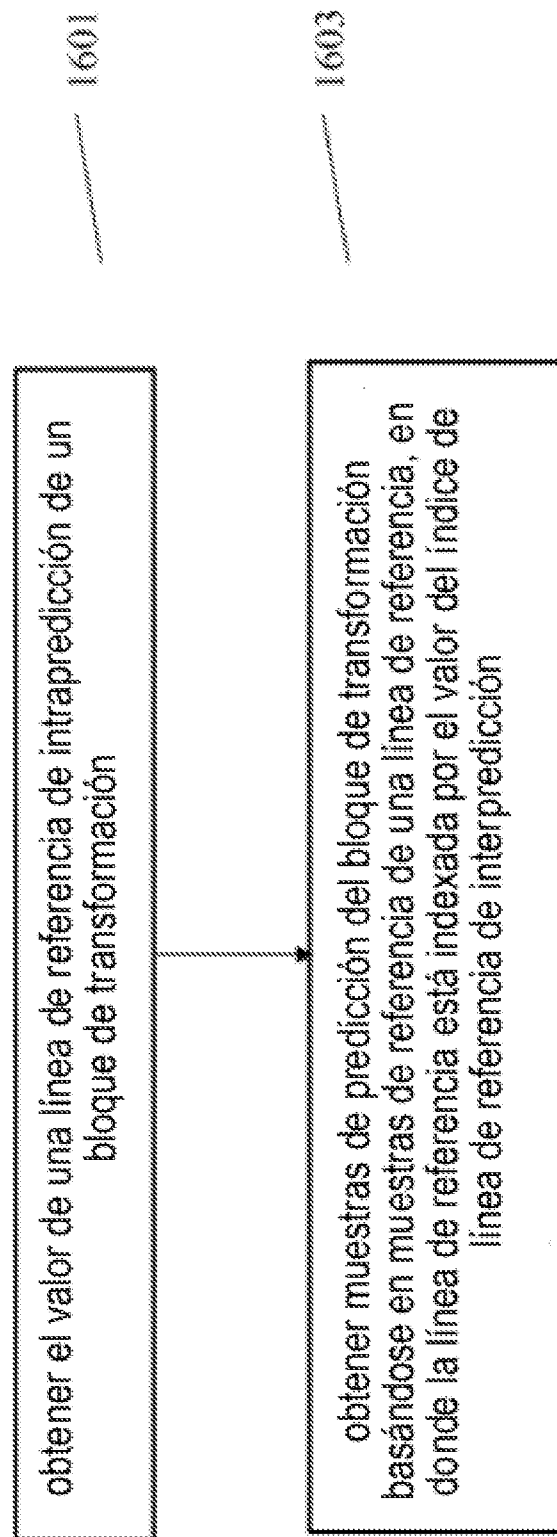


Fig. 14

20

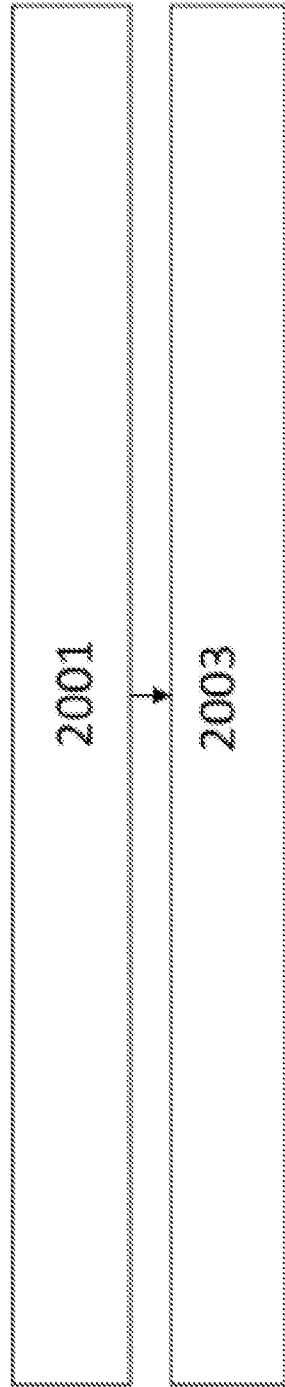


Fig. 15

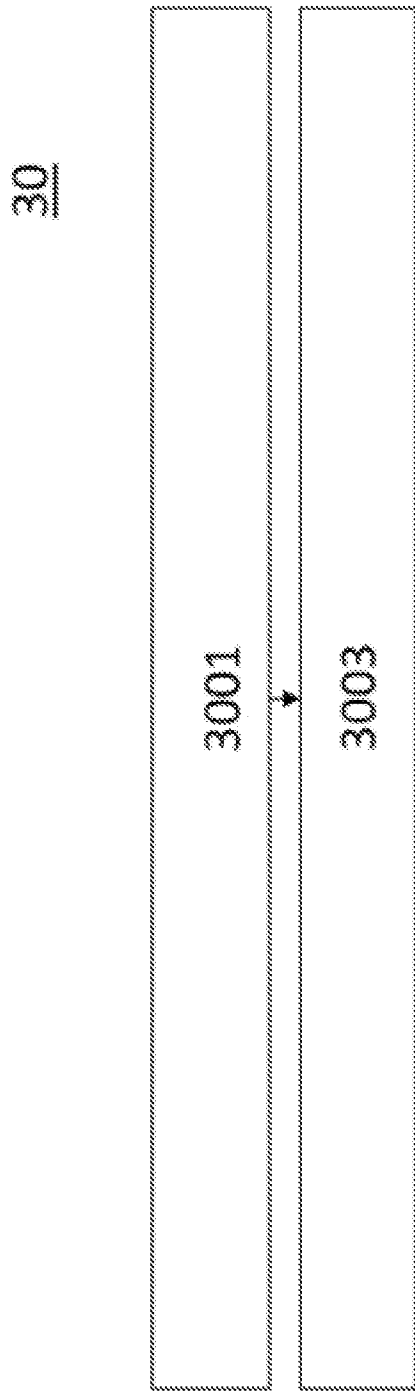


Fig. 16

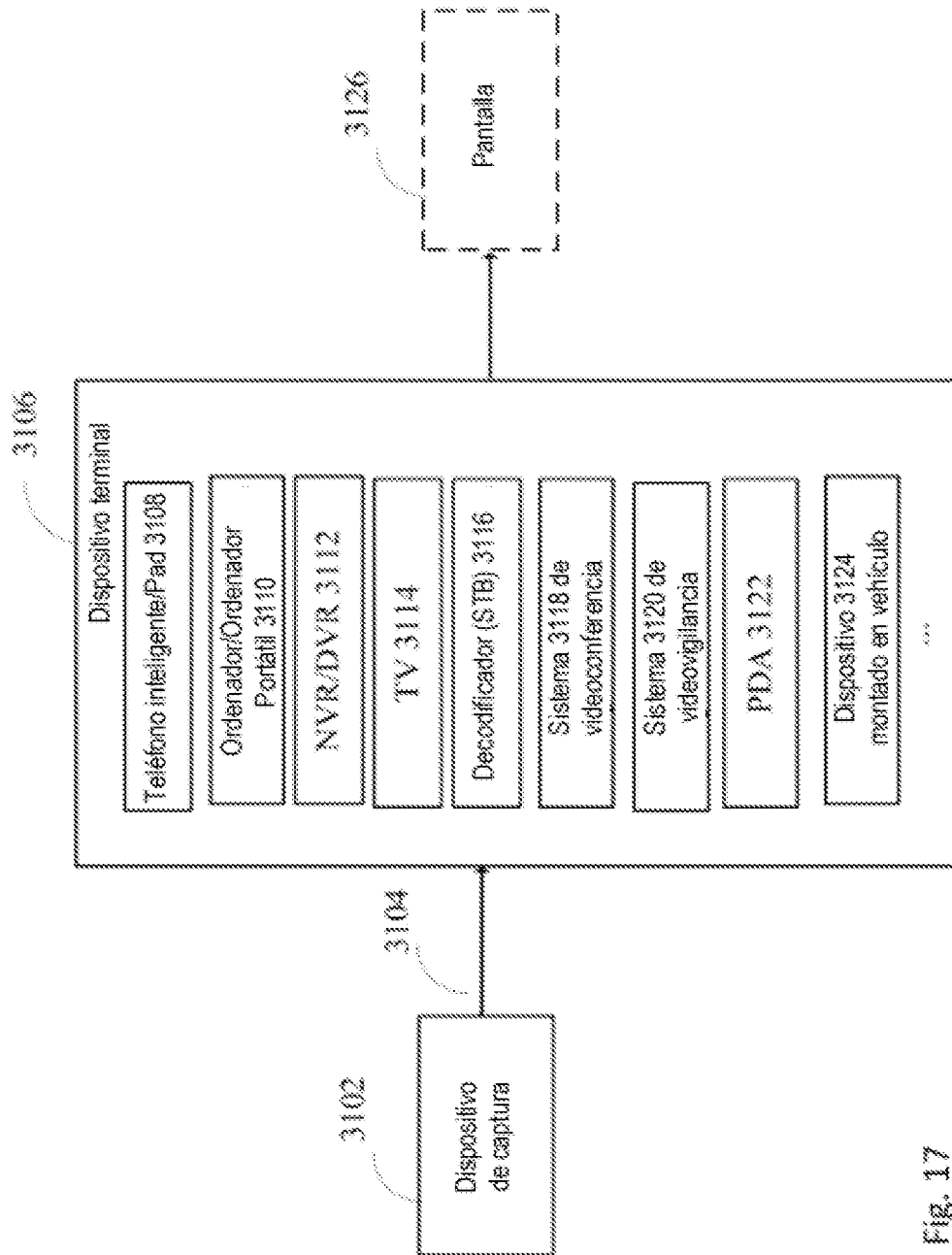


Fig. 17

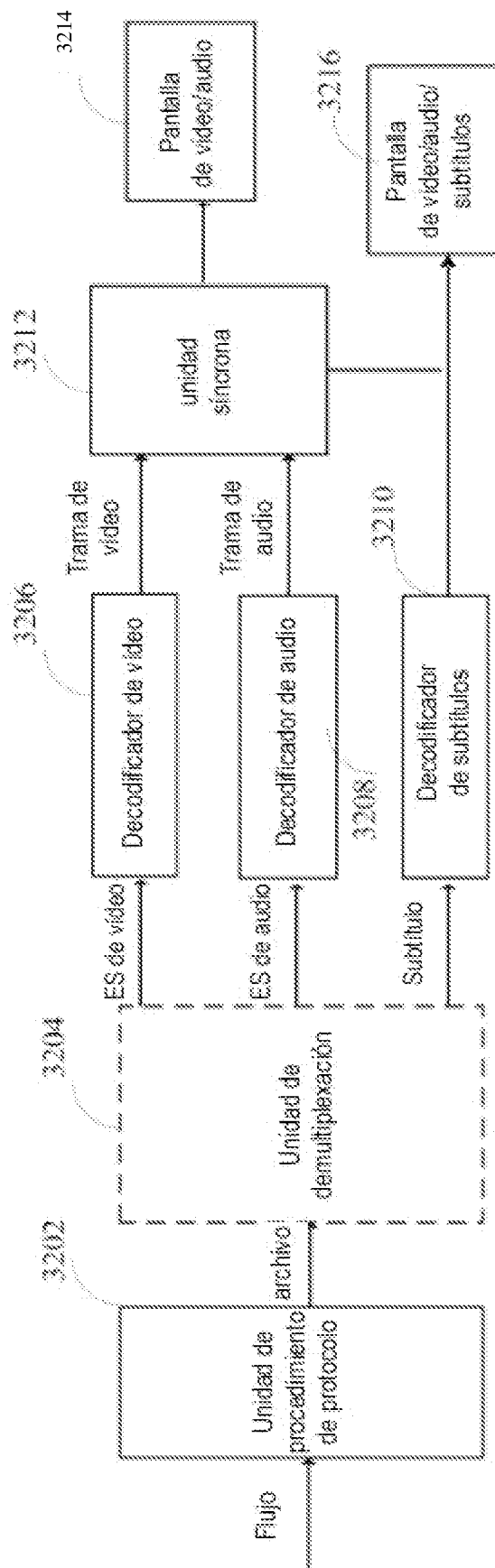


Fig. 18