

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 024 974**

51 Int. Cl.:

H04N 19/119 (2014.01)

H04N 19/176 (2014.01)

H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **19.06.2020** **PCT/CN2020/097032**

87 Fecha y número de publicación internacional: **24.12.2020** **WO20253816**

96 Fecha de presentación y número de la solicitud europea: **19.06.2020** **E 20825757 (6)**

97 Fecha y número de publicación de la concesión europea: **05.03.2025** **EP 3915256**

54 Título: **Un codificador, un decodificador y métodos correspondientes para el modo de particionamiento de subbloques**

30 Prioridad:

21.06.2019 WO PCT/EP2019/066432

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
05.06.2025

73 Titular/es:

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

72 Inventor/es:

ESENLIK, SEMIH;
BLAESER, MAX;
ZHAO, ZHIJIE;
GAO, HAN;
KOTRA, ANAND MEHER;
WANG, BIAO y
ALSHINA, ELENA ALEXANDROVNA

74 Agente/Representante:

PONS ARIÑO, Ángel

ES 3 024 974 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Un codificador, un decodificador y métodos correspondientes para el modo de particionamiento de subbloques

5 Campo técnico

Las realizaciones de la presente solicitud (invención) se relacionan en general con el campo del procesamiento de imágenes y más particularmente con la predicción utilizando modos de partición de subbloques.

10 Antecedentes

La codificación de vídeo (codificación y decodificación de vídeo) se utiliza 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 como video chat, videoconferencia, discos DVD y Blu-ray, sistemas de adquisición y edición de contenidos de vídeo y videocámaras para 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 porque los recursos de memoria pueden ser limitados. Los dispositivos de compresión de vídeo a menudo usan software y/o hardware en la fuente para codificar los datos de vídeo antes de la transmisión o el almacenamiento, lo que reduce la cantidad de datos necesarios para representar imágenes de vídeo digital. Luego, los datos comprimidos son recibidos en el destino por un dispositivo de descompresión de vídeo que decodifica los datos de vídeo. Con recursos de la red limitados y demandas cada vez mayores de mayor calidad de vídeo, son deseables técnicas mejoradas de compresión y descompresión que mejoren la relación de compresión con poco o ningún sacrificio en la calidad de imagen.

El documento EP2446627 A1 (Thomson Licensing, 2 de mayo de 2012) presenta métodos y aparatos para la codificación y decodificación de vídeo utilizando particionamiento geométrico adaptativo.

El documento EP2039171 A1 (Ericsson Telefon AB LM, 25 de marzo de 2009) presenta la codificación y decodificación de elementos de imagen de un fotograma en una secuencia de vídeo.

35 Compendio

La presente invención se define por las reivindicaciones adjuntas. Las realizaciones de la presente invención proporcionan aparatos y métodos para codificar y decodificar de acuerdo con las reivindicaciones independientes.

El objeto anterior se logra mediante las características de las reivindicaciones independientes. Otras formas de implementación se desprenden de las reivindicaciones dependientes, la descripción y las figuras.

De acuerdo con realizaciones de la presente invención, los parámetros para particionar un bloque (un parámetro de ángulo, un parámetro de distancia) se almacenan en una tabla de búsqueda predefinida, por lo tanto, los valores reales de estos parámetros no necesitan transmitirse en un flujo de bits, y los valores de estos parámetros se obtienen de acuerdo con un valor de indicación codificado en el flujo de bits. De esta forma se puede mejorar la eficiencia de la codificación.

Como se muestra en la Figura 15, se divulga un método de codificación implementado por un dispositivo de decodificación, el método comprende: S1501: obtener un flujo de bits.

El flujo de bits se puede obtener según la red inalámbrica o la red cableada. El flujo de bits puede transmitirse desde un sitio web, servidor u otra fuente remota mediante un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL) o tecnologías inalámbricas como infrarrojos, radio, microondas, WIFI, Bluetooth, LTE o 5G.

En una realización, un flujo de bits es una secuencia de bits, por ejemplo en forma de un flujo de unidad de capa de abstracción de red (NAL) o un flujo de bytes, que forma la representación de una secuencia de unidades de acceso (AU) que forman una o más secuencias de vídeo codificadas (CVS).

En algunas realizaciones, para un proceso de decodificación, un lado del decodificador lee un flujo de bits y deriva imágenes decodificadas del flujo de bits; para un proceso de codificación, un lado del codificador produce un flujo de bits.

Normalmente, un flujo de bits comprenderá elementos de sintaxis que están formados por una estructura de sintaxis. elemento de sintaxis: Un elemento de datos representado en el flujo de bits.

5 Estructura de sintaxis: Cero o más elementos de sintaxis presentes juntos en el flujo de bits en un orden especificado.

En un ejemplo específico, los formatos de flujo de bits especifican la relación entre el flujo de unidad de capa de abstracción de red (NAL) y el flujo de bytes, a cualquiera de los cuales se hace referencia como el flujo de bits.

10

El flujo de bits puede tener, por ejemplo, uno de dos formatos: el formato de flujo de unidad NAL o el formato de flujo de bytes. El formato de flujo de unidad NAL es conceptualmente el tipo más "básico". El formato de flujo de unidad NAL comprende una secuencia de estructuras de sintaxis llamadas unidades NAL. Esta secuencia está ordenada en orden de decodificación. Existen restricciones impuestas al orden de decodificación (y al contenido) de las unidades NAL en el flujo de unidades NAL.

15

El formato de flujo de bytes se puede construir a partir del formato de flujo de unidad NAL ordenando las unidades NAL en orden de decodificación y anteponiendo a cada unidad NAL un prefijo de código de inicio y cero o más bytes de valor cero para formar un flujo de bytes. El formato de flujo de unidad NAL se puede extraer del formato de flujo de bytes buscando la ubicación del patrón de prefijo de código de inicio único dentro de este flujo de bytes.

20

Esta cláusula especifica una realización de la relación entre las imágenes de origen y decodificadas que se proporciona a través del flujo de bits.

25

La fuente de vídeo que está representada por el flujo de bits es una secuencia de imágenes en orden de decodificación.

Las imágenes de origen y decodificadas se componen cada una de una o más matrices de muestra:

30

- Solo luma (Y) (monocromo).
- Luma y dos cromas (YCbCr o YCgCo).
- 35 - Verde, azul y rojo (GBR, también conocido como RGB).
- Matrices que representan otras muestras de color monocromáticas o triestímulo no especificadas (por ejemplo, YZX, también conocido como XYZ).

40

Las variables y términos asociados con estas matrices se denominan luma (o L o Y) y croma, donde las dos matrices de croma se denominan Cb y Cr; independientemente del método de representación de color real en uso. El método de representación de color real en uso se puede indicar en la sintaxis que se especifica en los parámetros VUI como se especifica en ITU-T H.261 | ISO/IEC 23002-7.

45

Las variables SubWidthC y SubHeightC se especifican en la tabla 2, dependiendo de la estructura de muestreo del formato cromático, que se especifica a través de sps_chroma_format_idc y sps_separate_colour_plane_flag.

50

Tabla 2: Valores SubWidthC y SubHeightC derivados de sps_chroma_format_idc y sps_separate_colour_plane_flag

sps_chroma_format_idc	sps_separate_colour_plane_flag	Formato cromático	SubWidthC	SubHeightC
0	0	Monocromo	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

En el muestreo monocromático solo hay una matriz de muestra, que nominalmente se considera la matriz de luma.

55

En el muestreo 4:2:0, cada una de las dos matrices de croma tiene la mitad de la altura y la mitad del ancho de la matriz de luma.

En el muestreo 4:2:2, cada una de las dos matrices de croma tiene la mitad de la altura y la mitad del ancho de la matriz de luma.

5 En el muestreo 4:4:4, dependiendo del valor de `sps_separate_colour_plane_flag`, se aplica lo siguiente:

- Si `sps_separate_colour_plane_flag` es igual a 0, cada una de las dos matrices de croma tiene la misma altura y ancho que la matriz de luma.

10 - De lo contrario (`sps_separate_colour_plane_flag` es igual a 1), los tres planos de color se procesan por separado como imágenes muestreadas monocromáticas.

S1502: obtener un valor de un indicador para un bloque actual de acuerdo con el flujo de bits.

15 En una realización, el valor del indicador se utiliza para especificar una forma de particionamiento del modo de fusión de particionamiento geométrico. Por ejemplo, el indicador puede ser `merge_gpm_partition_idx[x0][y0]`, en donde `merge_gpm_partition_idx[x0][y0]` especifica la forma de particionamiento del modo de fusión de particionamiento geométrico. 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.

Normalmente, el valor de `merge_gpm_partition_idx[x0][y0]` se decodifica a partir de un flujo de bits. En un ejemplo, un rango de valores para `merge_gpm_partition_idx[][]` es de 0 a 63, incluidos 0 y 63. En un ejemplo, un proceso de decodificación para `merge_gpm_partition_idx[][]` es "bypass".

25 Cuando `merge_gpm_partition_idx[x0][y0]` no está presente, se infiere que es igual a 0.

S1503: obtener un valor de un primer parámetro para el bloque actual y un valor de un segundo parámetro para el bloque actual, de acuerdo con el valor del indicador y una tabla de búsqueda predefinida.

30 En una implementación, el primer parámetro representa un ángulo (o angulares) para el particionamiento del bloque actual.

35 En una implementación, el segundo parámetro representa una distancia para el particionamiento del bloque actual.

40 En una implementación, la tabla de búsqueda predefinida comprende pares de parámetros primero y segundo, en donde cuando el primer parámetro de un par representa un ángulo correspondiente a uno de los ángulos de 0, 45, 90, 135, 180, 225, 270 o 315 grados, el segundo parámetro correspondiente del par no representa una distancia de cero muestras desde el centro del bloque actual.

45 En una implementación, la tabla de búsqueda predefinida comprende pares de parámetros primero y segundo, en donde cuando el primer parámetro de un par representa un ángulo no correspondiente a uno de los ángulos de 0, 45, 90, 135, 180, 225, 270 o 315 grados, y el segundo parámetro correspondiente del par representa una distancia de cero muestras desde el centro del bloque actual.

50 En una implementación, la tabla de búsqueda predefinida comprende pares de parámetros primero y segundo, en donde cuando el primer parámetro de un par representa un ángulo correspondiente a uno de los ángulos de 0, 45, 90, 135, 180, 225, 270 o 315 grados, y el segundo parámetro correspondiente del par representa una distancia de cero muestras desde el centro del bloque actual.

55 En una implementación, en donde el segundo parámetro especifica un índice de distancia de una partición geométrica, o el segundo parámetro describe una distancia de una línea de separación al centro del bloque actual.

60 En un ejemplo, la variable de ángulo de partición `angleIdx` (parámetro 1) y la variable de distancia `distanceldx` (parámetro 2) del modo de particionamiento geométrico se establecen de acuerdo con el valor de `merge_gpm_partition_idx[xCb][yCb]` (indicador) como se especifica en la siguiente tabla. Se podría entender que, en la implementación, esta relación se puede implementar de acuerdo con la tabla 1 o de acuerdo con una función.

Tabla 1: Especificación de `angleIdx` y `distanceldx` según `merge_gpm_partition_idx`.

<code>merge_gpm_partition_idx</code>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<code>angleIdx</code>	0	0	2	2	2	2	3	3	3	3	4	4	4	4	5	5

merge_gpm_partition_idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
distanceldx	1	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1
merge_gpm_partition_idx	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
anglidx	5	5	8	8	11	11	11	11	12	12	12	12	13	13	13	13
distanceldx	2	3	1	3	0	1	2	3	0	1	2	3	0	1	2	3
merge_gpm_partition_idx	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
anglidx	14	14	14	14	16	16	18	18	18	19	19	19	20	20	20	21
distanceldx	0	1	2	3	1	3	1	2	3	1	2	3	1	2	3	1
merge_gpm_partition_idx	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
anglidx	21	21	24	24	27	27	27	28	28	28	29	29	29	30	30	30
distanceldx	2	3	1	3	1	2	3	1	2	3	1	2	3	1	2	3

S1504: obtener un valor de una distancia de muestra para una muestra que se encuentra en el bloque actual, de acuerdo con el valor del primer parámetro y el valor del segundo parámetro.

5 En una realización, este paso comprende:

10 Paso 3.1: obtener un valor de índice de un parámetro de ángulo (alphaN o anglidx) para el bloque actual, un valor de ancho del bloque actual (W), un valor de altura del bloque actual (H). W y H son el ancho y la altura del bloque actual en número de muestras. Por ejemplo, un bloque de codificación con ancho y alto iguales a 8 es un bloque cuadrado que comprende 64 muestras. En otro ejemplo, W y H son el ancho y la altura del bloque actual, en número de muestras de luma. En un ejemplo, el valor del índice del parámetro de ángulo se puede obtener de acuerdo con la descripción anterior haciendo referencia a la tabla 1.

15 Paso 3.2: obtener un valor de una relación whRatio en función del valor de W y el valor de H, representando el valor de whRatio una relación entre el ancho y la altura del bloque de codificación actual.

En un ejemplo, whRatio = H / W; o whRatio = W/H.

20 En otro ejemplo, dos variables nCbW y nCbH especifican el ancho y la altura del bloque de codificación actual, y una variable cIdx especifica un índice de componente de color. Las variables nW, nH y whRatio se derivan de la siguiente manera:

$$nW = (cIdx == 0) ? nCbW : nCbW * SubWidthC;$$

$$25 \quad nH = (cIdx == 0) ? nCbH : nCbH * SubHeightC;$$

$$whRatio = nH / nW.$$

30 En un ejemplo, las variables SubWidthC y SubHeightC se especifican en la tabla 2.

35 Paso 3.3: Obtener un valor shiftHor de acuerdo con una tabla de búsqueda, el valor de alfa y el valor de whRatio, en un ejemplo, el valor de alfa (alphaN o anglidx) y el valor de whRatio se utilizan como valores de índice de la tabla de búsqueda. El valor shiftHor también se puede obtener de acuerdo con una función, en donde el valor de alfa (alphaN o anglidx) y el valor de whRatio son la entrada de la función y el valor shiftHor es la salida de la función. En un ejemplo, el resultado de la función es similar o igual a la tabla de búsqueda. En un ejemplo, el valor shiftHor representa un tamaño de paso de cuantificación para el proceso de cálculo de la distancia de muestra.

40 En otro ejemplo, la función puede representarse por $shiftHor = (anglidx \% 16 == 8 \parallel (anglidx \% 16 != 0 \&\& whRatio > 0)) ? 0 : 1$.

Paso 3.4: se calcula un valor de sample_dist según el valor shiftHor.

45 En un ejemplo, para calcular la distancia de muestra (sample_dist), primero, las variables offsetX y offsetY se derivan de la siguiente manera: Si shiftHor es igual a 0, se aplica lo siguiente:

$$offsetX = (-nW) >> 1,$$

offsetY = ((-nH) >> 1) +

(angleIdx < 16 ? (distanceIdx * nH) >> 3 : -((distanceIdx * nH) >> 3));

De lo contrario (shiftHor es igual a 1), se aplica lo siguiente:

5 offsetY = ((-nW) >> 1) +

(angleIdx < 16 ? (distanceIdx * nW) >> 3 : -((distanceIdx * nW) >> 3));

10 offsetY = (- nH) >> 1;

- Las variables xL e yL se derivan de la siguiente manera:

xL = (cIdx == 0) ? x : x * SubWidthC

15 yL = (cIdx == 0) ? y : y * SubHeightC

sample_dist = (((xL + offsetX) << 1) + 1) * disLut[displacementX] +

(((yL + offsetY) << 1) + 1) * disLut[displacementY].

Las variables displacementX y displacementY se derivan de la siguiente manera:

20 hwRatio = cbHeight / cbWidth;

displacementX = angleIdx;

25 displacementY = (angleIdx + 8) % 32.

La matriz disLut se especifica en la siguiente tabla 3.

30 En algunas realizaciones, de acuerdo con un modelo geométrico, las muestras en un bloque de codificación se consideran ubicadas en dos subbloques. El subbloque A o el subbloque B pueden comprender una parte (pero no todas) de las muestras del bloque de codificación actual. El subbloque A o el subbloque B se pueden representar de acuerdo con el signo de una sample_dist de cada muestra. La sample_dist se puede obtener de acuerdo con los ejemplos y realizaciones de los otros párrafos.

35 S1505: obtener un valor de predicción para la muestra, de acuerdo con el valor de la distancia de muestra para la muestra.

En una implementación, la obtención de un valor de predicción para la muestra, de acuerdo con el valor de la distancia de muestra para la muestra, comprende:

40 calcular dos factores de ponderación en función del valor de la distancia de la muestra; y

obtener el valor de predicción para la muestra de acuerdo con un primer valor de predicción, un segundo valor de predicción y los dos factores de ponderación.

45 En una implementación, el valor de la distancia de muestra representa la distancia horizontal o la distancia vertical, o una combinación de las distancias verticales y horizontales, de dicha muestra a una línea de separación, en donde la línea de separación se utiliza para dividir un bloque de codificación en dos subbloques.

50 En un ejemplo, la sample_dist calculada se utiliza para calcular factores de ponderación, los factores de ponderación se utilizan para la combinación de un primer valor de predicción y un segundo valor de predicción correspondiente a dicha muestra. En un ejemplo, los factores de ponderación se denominan sampleWeight1 y sampleWeight2, haciendo referencia al peso correspondiente al primer valor de predicción y al peso correspondiente al segundo valor de predicción.

55 En un ejemplo, los factores de ponderación se calculan de acuerdo con las siguientes funciones,

weightIdxL = partFlip ? 32 + sample_dist : 32 - sample_dist;

$wValue = Clip3(0, 8, (weightIdxL + 4) \gg 3).$

En este ejemplo, wValue es sampleWeight1 y 8-wValue es sampleWeight2. La variable partFlip se determina según un valor de angleIdx. En un ejemplo, $partFlip = (angleIdx \geq 13 \ \&\& \ angleIdx \leq 27) ? 0 : 1$, o $partFlip = (angleIdx \geq 13 \ \&\& \ angleIdx \leq 27) ? 1 : 0$.

En un ejemplo, el valor combinado de la muestra de predicción en la coordenada de muestra (x,y) se calcula de acuerdo con el primer valor de predicción en la coordenada (x,y), el segundo valor de predicción en la coordenada (x,y), el sampleWeight1 y el sampleWeight2.

En un ejemplo, el valor de muestra de predicción se deriva de la siguiente manera:

$pbSamples[x][y] = Clip3(0, (1 \ll BitDepth) - 1, (predSamplesLA[x][y] * wValue +$

En donde bitDepth representa la profundidad de bits de la muestra, la variable shift1 se obtiene de acuerdo con bitDepth, en un ejemplo, $shift1 = \text{Max}(5, 17 - BitDepth)$; la variable offset1 se obtiene de acuerdo con shift1, en un ejemplo, $offset1 = 1 \ll shift1 - 1$, predSamplesLA y predSamplesLB son dos matrices (nCbw)x(nCbH).

En una realización, se describe un método de codificación implementado por un dispositivo de codificación, comprendiendo el método: seleccionar un valor de un primer parámetro y un valor de un segundo parámetro; obtener un valor de índice de acuerdo con el valor del primer parámetro, el valor del segundo parámetro y una tabla de búsqueda; y codificar el valor de índice en un flujo de bits.

Los detalles de cada paso en el lado del codificador corresponden a los ejemplos anteriores en el lado del decodificador.

Como se muestra en la Figura16, el segundo aspecto de la presente invención proporciona un dispositivo de decodificación 1600, comprendiendo el dispositivo de decodificación:

un módulo receptor 1601, que está configurado para obtener un flujo de bits y obtener un valor de un indicador para un bloque actual de acuerdo con el flujo de bits;

un módulo de procesamiento de parámetros de partición 1602, que está configurado para obtener un valor de un primer parámetro para el bloque actual y un valor de un segundo parámetro para el bloque actual, de acuerdo con el valor del indicador y una tabla de búsqueda predefinida;

un módulo de cálculo 1603, que está configurado para obtener un valor de una distancia de muestra para una muestra que se encuentra en el bloque actual, de acuerdo con el valor del primer parámetro y el valor del segundo parámetro; y un módulo de predicción 1604, que está configurado para obtener un valor de predicción para la muestra, de acuerdo con el valor de la distancia de muestra para la muestra.

El método según el primer aspecto de la invención puede llevarse a cabo mediante el aparato según el segundo aspecto de la invención. Otras características y formas de implementación de los métodos anteriores corresponden a las características y formas de implementación del aparato según el segundo aspecto de la invención.

En una realización, se describe un decodificador (30) o un codificador (20) que comprende circuitos de procesamiento para llevar a cabo el método de acuerdo con una cualquiera de las realizaciones e implementaciones anteriores.

En una realización, se divulga un producto de programa informático que comprende un código de programa para realizar el método de acuerdo con cualquiera de las realizaciones e implementaciones anteriores.

En una realización, un decodificador o un codificador, que comprende:

uno o más procesadores; y

En un ejemplo, se describe un medio 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 preprocesadores, configura el codificador o el decodificador para llevar a cabo el método de acuerdo con cualquiera de las realizaciones e implementaciones anteriores. En un ejemplo, un medio de almacenamiento no transitorio que incluye un flujo de bits codificado decodificado por un dispositivo de decodificación de imágenes, el flujo de bits se genera dividiendo un fotograma de una señal de

vídeo o una señal de imagen en una pluralidad de bloques, e incluye una pluralidad de elementos de sintaxis, en donde la pluralidad de elementos de sintaxis comprende un indicador (sintaxis) de acuerdo con cualquiera de las realizaciones e implementación anteriores.

- 5 Los detalles de una o más realizaciones se exponen en los dibujos que se acompaña y la descripción que sigue. Otras características, objetos y ventajas serán evidentes en función de la descripción, los dibujos y las reivindicaciones.

Breve descripción de los dibujos

- 10 A continuación se describen con más detalle realizaciones de la invención con referencia a las figuras y dibujos adjuntos, en los que:

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

- 20 la Figura 2 es un diagrama de bloques que muestra un ejemplo de un codificador de vídeo configurado para implementar realizaciones de la invención;

la Figura 3 es un diagrama de bloques que muestra una estructura de ejemplo de un codificador de vídeo configurado para implementar realizaciones de la invención;

- 25 la Figura 4 es un diagrama de bloques que ilustra un ejemplo de un aparato de codificación o un aparato de decodificación;

- 30 la Figura 5 es un diagrama de bloques que ilustra otro ejemplo de un aparato de codificación o un aparato de decodificación;

la Figura 6a se ilustra un ejemplo de bloque cúbico;

la Figura 6b ilustra un ejemplo de bloques vecinos espaciales;

- 35 la Figura 7 ilustra algunos ejemplos del modo de predicción triangular;

la Figura 8 ilustra algunos ejemplos del modo de predicción de subbloques;

- 40 las Figuras 9 a 12 muestran algunos ejemplos sobre la partición de un bloque;

la Figura 13 es un diagrama de bloques que muestra una estructura de ejemplo de un sistema de suministro de contenido 3100 que realiza un servicio de suministro de contenidos;

- 45 la Figura 14 es un diagrama de bloques que muestra la estructura de un ejemplo de un dispositivo terminal;

la Figura 15 es un diagrama de flujo que muestra una realización de un método de acuerdo con la presente invención.

- 50 la Figura 16 es un diagrama de bloques que muestra una realización de un aparato de acuerdo con la presente invención.

En lo que sigue, los signos de referencia idénticos se refieren a características idénticas o al menos funcionalmente equivalentes, a menos que se especifique explícitamente lo contrario.

- 55 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 limitado, y el alcance de la presente invención se define mediante las reivindicaciones adjuntas.

- 65 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 específicas del método, un dispositivo correspondiente puede incluir una o una pluralidad de unidades, por ejemplo, unidades funcionales, para realizar una o la pluralidad de etapas del método descritas (por ejemplo, una unidad que realiza una o la pluralidad de etapas, o una pluralidad de unidades, cada una de las cuales realiza uno o más de la pluralidad de etapas), incluso si dichas
5 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 una o la pluralidad de unidades (por ejemplo, una etapa que realiza la funcionalidad de 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
10 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.

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 "fotograma" o "imagen" se puede
15 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 necesarios para representar las imágenes de vídeo (para un almacenamiento y/o transmisión más eficiente). La decodificación de vídeo se
20 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 secuencias de vídeo. La combinación de la parte de codificación y la parte de decodificación también se conoce como CODEC (codificación y decodificación).
25

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
30 caso de codificación de vídeo con pérdida, 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 a la calidad de las imágenes de vídeo originales.

Varios estándares de codificación de vídeo pertenecen al grupo de "códecs de vídeo híbridos con pérdida" (es decir, combinan predicción espacial y temporal en el dominio de muestra y codificación de transformación 2D para aplicar cuantificación en el dominio de 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,
40 se codifica, en un nivel de bloque (bloque de vídeo), por ejemplo, usando predicción espacial (intraimagen) y 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 parcialmente al bloque codificado o comprimido el procesamiento inverso al del codificador para reconstruir el bloque actual para su representación. Además, el codificador duplica el bucle de procesamiento del decodificador de manera que
45 ambos generarán predicciones idénticas (por ejemplo, intrapredicciones e interpredicciones) y/o reconstrucciones para el procesamiento, es decir, la codificación, de los bloques subsiguientes.

En las siguientes realizaciones de un sistema de codificación de vídeo 10, se describen un codificador de vídeo 20 y un decodificador de vídeo 30 con base en las figuras 1 a 3.

La Figura 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 sistema 10 de codificación corta) que puede utilizar
55 técnicas de esta presente invención. El codificador de vídeo 20 (o codificador corto 20) y el decodificador de vídeo 30 (o decodificador corto 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 invención.

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

El dispositivo 12 de origen comprende un codificador 20, y puede comprender adicionalmente, es decir,
65 opcionalmente, un origen 16 de imágenes, un preprocesador (o unidad de preprocesamiento) 18, por ejemplo, un preprocesador de imágenes 18, 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 dispositivo de generación de imágenes, por ejemplo un procesador de gráficos por ordenador para generar una imagen animada por ordenador, o cualquier otro tipo de dispositivo para obtener y/o proporcionar una imagen del mundo real, una imagen animada 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)). La fuente de la imagen puede ser cualquier tipo de memoria o almacenamiento que almacene cualquiera de las imágenes mencionadas anteriormente.
- 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.
- El preprocesador 18 está configurada 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.
- El codificador de vídeo 20 está configurado para recibir los datos 19 de imágenes preprocesados y proporcionar datos 21 de imágenes codificados (a continuación se describirán más detalles, por ejemplo, basándose en la Figura 2).
- La interfaz 22 de comunicación del dispositivo de origen 12 puede configurarse para recibir los datos 21 de imágenes codificados y para transmitir los datos 21 de imágenes codificados (o cualquier versión procesada adicional de los mismos) a través del canal de comunicación 13 a otro dispositivo, por ejemplo el dispositivo de destino 14 o cualquier otro dispositivo, para almacenamiento o reconstrucción directa.
- El dispositivo 14 de destino comprende un decodificador 30 (por ejemplo, un decodificador de vídeo 30), y puede comprender adicionalmente, es decir, opcionalmente, una interfaz de comunicación o unidad 28 de comunicación, un postprocesador 32 (o una unidad 32 de posprocesamiento) y un dispositivo 34 de visualización.
- La interfaz 28 de comunicación del dispositivo de destino 14 está configurada para recibir los datos 21 de imágenes codificados (o cualquier versión procesada adicional de los mismos), por ejemplo directamente desde el dispositivo 12 de origen o desde cualquier otra fuente, por ejemplo un dispositivo de almacenamiento, por ejemplo, un dispositivo de almacenamiento de datos de imágenes codificados, y proporcionar los datos 21 de imágenes codificados al decodificador 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. La interfaz 22 de comunicación puede estar configurada, por ejemplo, para empaquetar los datos 21 de imágenes codificados en un formato apropiado, por ejemplo, paquetes, y/o procesar los datos de imágenes codificados utilizando cualquier tipo de codificación o procesamiento de transmisión para su transmisión a través de un enlace de comunicación o red de comunicación.
- La interfaz 28 de comunicación, que forma la contraparte de la interfaz 22 de comunicación, puede estar, por ejemplo, configurada para recibir los datos transmitidos y procesar los datos de transmisión utilizando cualquier tipo de decodificación o procesamiento de transmisión correspondiente y/o desempaquetado para obtener los datos de imágenes codificados 21.
- 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 de comunicación 13 en la Figura 1A que apunta 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 acusar 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.
- El decodificador 30 está configurado para recibir los datos 21 de imágenes codificados y proporcionar datos 31 de imágenes decodificados o una imagen 31 decodificada (a continuación se describirán más detalles, por ejemplo, basándose en la Figura 3 o la Figura 5).

El postprocesador 32 del dispositivo 14 de destino está configurado para postprocesar los datos 31 de imágenes decodificados (también llamados datos de imágenes reconstruidos), por ejemplo, la imagen 31 decodificada, para obtener datos 33 de imágenes postprocesados, por ejemplo, una imagen 33 postprocesada. El postprocesamiento realizado por la unidad 32 de postprocesamiento 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 imágenes decodificados para la visualización, por ejemplo, mediante el dispositivo 34 de visualización.

El dispositivo 34 de visualización del dispositivo 14 de destino está configurado para recibir los datos 33 de imágenes posprocesados para mostrar 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 Figura 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 por hardware y/o software separados o cualquier combinación de los mismos.

Como será evidente para el experto en la técnica en función de 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 Figura 1A, puede variar dependiendo del dispositivo y la aplicación reales.

El codificador 20 (por ejemplo, un codificador de vídeo 20) o el decodificador 30 (por ejemplo, un decodificador de vídeo 30) o tanto el codificador 20 como el decodificador 30 pueden implementarse mediante circuitos de procesamiento como se muestra en la Figura 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, codificación de vídeo dedicada o cualquier combinación de los mismos. El codificador 20 puede implementarse a través de un circuito de procesamiento 46 para incorporar los diversos módulos como se analiza con respecto al codificador 20 de la Figura 2 y/o cualquier otro sistema o subsistema codificador descrito en esta memoria. El decodificador 30 puede implementarse a través de un circuito de procesamiento 46 para incorporar los diversos módulos como se analiza con respecto al codificador 30 de la Figura 3 y/o cualquier otro sistema o subsistema codificador descrito en esta memoria. Los circuitos de procesamiento pueden configurarse para realizar las distintas operaciones que se analizarán más adelante. Como se muestra en la Figura 5, si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio de almacenamiento legible por ordenador no transitorio adecuado y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta invención. Cualquiera del codificador de vídeo 20 o del decodificador de vídeo 30 pueden integrarse como parte de un codificador/decodificador combinado (CODEC) en un solo dispositivo, por ejemplo, como se muestra en la Figura 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, portátiles u ordenadores portátiles, teléfonos móviles, teléfonos inteligentes, tabletas u ordenadores de tableta, cámaras, ordenadores de escritorio, decodificadores, televisores, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de vídeo (tales como servidores de servicios de contenido o servidores de suministro de contenido), dispositivo receptor de emisiones, dispositivo transmisor de emisiones o similares, y puede usar ninguno 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 Figura 1A es simplemente un ejemplo y las técnicas de la presente invención pueden aplicarse a configuraciones de codificación de vídeo (por ejemplo, codificación de vídeo o decodificación de vídeo) que no necesariamente incluyen alguna 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.

- Para facilitar la descripción, en esta memoria se describen realizaciones de la invención, por ejemplo, en referencia a la codificación de vídeo de alta eficiencia (HEVC) o al software de referencia de codificación de vídeo versátil (VVC), desarrollado por el equipo de colaboración conjunta en codificación de vídeo (JCT-VC) del grupo de expertos en codificación de vídeo del ITU-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

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

- La unidad de cálculo residual 204, la unidad de procesamiento de transformada 206, la unidad de cuantificación 208, la unidad de selección de modo 260 pueden ser referidas como formadoras de una ruta de señal directa del codificador 20, mientras que la unidad de cuantificación inversa 210, la unidad de procesamiento de transformada inversa 212, la unidad de reconstrucción 214, el búfer 216, el filtro de bucle 220, el búfer de imagen decodificada (DPB) 230, la unidad de interpredicción 244 y la unidad de intrapredicción 254 pueden ser referidas como formadoras de una ruta de señal hacia atrás del codificador de vídeo 20, en donde la ruta de señal hacia atrás del codificador de vídeo 20 corresponde a la ruta de señal del decodificador (véase el decodificador de vídeo 30 en la Figura 3). La unidad 210 de cuantificación inversa, la unidad 212 de procesamiento de transformación inversa, la unidad de reconstrucción 214, el filtro de bucle 220, el búfer de imagen decodificada (DPB) 230, la unidad de interpredicción 244 y la unidad de intrapredicción 254 también se denominan "decodificador incorporado" del codificador de vídeo 20.

- Imágenes y particiones de imágenes (imágenes y bloques)

- El codificador 20 puede estar configurado para recibir, por ejemplo a través de la entrada 201, una imagen 17 (o datos de imagen 17), por ejemplo una imagen de una secuencia de imágenes que forman un vídeo o una secuencia de vídeo. La imagen o los datos de imágenes recibidos también pueden ser una imagen preprocesada 19 (o datos de imágenes preprocesados 19). Para simplificar, la siguiente descripción se refiere a la imagen 17. La imagen 17 también puede denominarse como imagen actual o 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).

- 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 como píxel (forma abreviada de elemento de imagen) o 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 muestra 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 conoce como 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 formato de color 4:2:0, 4:2:2 y 4:4:4.

Las realizaciones del codificador de vídeo 20 pueden comprender una unidad de particionamiento (no representada en la Figura 2) configurada para particionar la imagen 17 en una pluralidad de bloques 203 de imágenes (normalmente no superpuestos). Estos bloques también pueden denominarse bloques raíz, macrobloques (H.264/AVC) o bloques de árbol de codificación (CTB) o unidades de árbol de codificación (CTU) (H.265/HEVC y VVC). La unidad de particionamiento de imágenes 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.

En otras realizaciones, el codificador de vídeo puede estar configurado 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.

Al igual que la imagen 17, el bloque 203 de imágenes 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 el caso de una imagen en 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 $M \times N$ (M columnas por N filas) de muestras, o una matriz $M \times N$ de coeficientes de transformación.

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

Cálculo residual

La unidad 204 de cálculo residual puede estar configurada para calcular un bloque 205 residual (también denominado residual 205) en función del bloque 203 de imágenes 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 imágenes, muestra por muestra (píxel por píxel) para obtener el bloque 205 residual en el dominio de la muestra.

Transformada

La unidad 206 de procesamiento de transformada puede estar configurada para aplicar una transformada, 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 transformada en un dominio de transformada. Los coeficientes 207 de transformada también pueden denominarse coeficientes residuales de transformada y representan el bloque 205 residual en el dominio de transformada.

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

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

Cuantificación

La unidad 208 de cuantificación está configurada para cuantificar los coeficientes 207 de transformada para

obtener coeficientes 209 cuantificados, por ejemplo, aplicando cuantificación escalar o cuantificación vectorial. Los coeficientes 209 cuantificados también pueden denominarse coeficientes 209 de transformada cuantificados o coeficientes 209 residuales cuantificados.

- 5 El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes 207 de transformada. Por ejemplo, un coeficiente de transformada de n bits puede redondearse hacia abajo a un coeficiente de transformada de m bits durante la cuantización, donde n es mayor que m . El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación (QP). Por ejemplo, para la
- 10 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
- 15 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 la descuantificación correspondiente o inversa, por ejemplo, mediante la cuantificación 210 inversa, puede incluir la multiplicación por el tamaño del escalón de
- 20 cuantificación. Los ejemplos 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 en función de un parámetro de cuantificación usando una aproximación de punto fijo de una ecuación que incluye la división. Pueden introducirse factores de escala adicionales para la cuantificación y la descuantificación 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
- 25 de cuantificación y el parámetro de cuantificación. En una implementación de ejemplo, se pueden combinar el escalado de la transformada inversa y la descuantificación. 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 la que la pérdida aumenta al aumentar los tamaños de
- 30 los escalones de cuantificación.

- Las realizaciones del codificador de vídeo 20 (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, de modo que, por ejemplo, el decodificador de vídeo 30 pueda
- 35 recibir y aplicar los parámetros de cuantificación para la decodificación.

Cuantificación inversa

- La unidad 210 de cuantificación inversa está configurada para aplicar la cuantificación inversa de la unidad 208
- 40 de cuantificación en los coeficientes cuantificados para obtener los coeficientes 211 descuantificados, por ejemplo, aplicando el esquema de cuantificación inversa aplicado por la unidad 208 de cuantificación en función de o usando el mismo tamaño del escalón de cuantificación que la unidad 208 de cuantificación. Los coeficientes 211 descuantificados también pueden denominarse coeficientes 211 residuales descuantificados y corresponden - aunque generalmente no son idénticos a los coeficientes de transformada debido a la pérdida por cuantificación - a los coeficientes 207 de transformada.
- 45

Transformada inversa

- La unidad 212 de procesamiento de transformada inversa está configurada para aplicar la transformada inversa de la transformada aplicada por la unidad 206 de procesamiento de transformada, por ejemplo, una transformación de coseno discreta inversa (DCT) o una transformación de seno discreta inversa (DST), u otras transformadas inversas para obtener un bloque 213 residual reconstruido (o coeficientes 213 descuantificados correspondientes) en el dominio de la muestra. El bloque 213 residual reconstruido también puede denominarse bloque 213 de transformada.
- 50

Reconstrucción

- La unidad 214 de reconstrucción (por ejemplo, sumador 214) está configurada para añadir el bloque 213 de transformada (es decir, el bloque 213 residual reconstruido) al bloque 265 de predicción para obtener un bloque
- 60 215 reconstruido en el dominio de la muestra, por ejemplo, añadiendo 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 "filtro de bucle" 220 para abreviar), está configurada para filtrar el bloque 215 reconstruido para obtener un bloque filtrado 221, o en general, para filtrar muestras reconstruidas para

obtener muestras filtradas. La unidad de filtro de bucle está configurada, por ejemplo, para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del 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) u otros filtros, por ejemplo, un filtro bilateral, un filtro de bucle adaptativo (ALF),
 5 filtros de nitidez, filtros de suavizado o filtros colaborativos. o cualquier combinación de los mismos. Aunque la unidad 220 de filtro de bucle se muestra en la Figura 2 como un filtro de bucle, en otras configuraciones, la unidad 220 de filtro de bucle puede implementarse como un filtro posterior al bucle. El bloque 221 filtrado también puede denominarse bloque 221 filtrado reconstruido.

10 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 en entropía 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.

15 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 imágenes de referencia, para la codificación de datos de vídeo mediante el codificador de vídeo 20. El DPB 230 puede estar formado por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluida la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El búfer de imágenes decodificadas (DPB) 230 se puede configurar para almacenar uno o más bloques filtrados 221. El búfer de imágenes decodificadas 230 puede configurarse además para almacenar otros bloques previamente
 20 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 interpolación. El búfer de imagen decodificada (DPB) 230 también puede
 25 configurarse para almacenar uno o más bloques 215 reconstruidos sin filtrar, o en general muestras reconstruidas sin filtrar, por ejemplo si el bloque 215 reconstruido no es filtrado por la unidad 220 de filtro de bucle, o cualquier otra versión procesada adicionalmente de los bloques o muestras reconstruidos.

35 Selección de modo (particionamiento y predicción)

La unidad 260 de selección de modo comprende una unidad de particionamiento 262, una unidad de interpolación 244 y una unidad de intrapredicción 254, y está configurada para recibir u obtener datos de imagen original, por ejemplo, un bloque 203 original (bloque actual 203 de la imagen actual 17), y datos de imagen reconstruida, por ejemplo, muestras o bloques reconstruidos filtrados y/o sin filtrar de la misma imagen
 40 (actual) y/o de una o una pluralidad de imágenes decodificadas previamente, por ejemplo, del búfer de imagen decodificada 230 u otros búferes (por ejemplo, búfer de línea, no mostrado). Los datos de imagen reconstruidos se utilizan como datos de imagen de referencia para la predicción, por ejemplo, interpolación 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 un particionamiento para un modo de predicción de
 45 bloque actual (incluida ninguna partición) y un modo de predicción (por ejemplo, un modo de intrapredicción o interpolación) y generar un bloque 265 de predicción correspondiente, que se utiliza 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 el
 50 particionamiento y el modo de predicción (por ejemplo, de aquellos soportados por 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
 55 para determinar el particionamiento y el modo de predicción en función de 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" general, etc. sino que también pueden referirse al cumplimiento de un criterio de terminación o selección como un valor que excede o cae por debajo de un umbral u otras restricciones que conducen
 60 potencialmente a una "selección subóptima" pero que reducen la complejidad y el tiempo de procesamiento.

En otras palabras, la unidad de particionamiento 262 puede configurarse para particionar el bloque 203 en particiones de bloque o subbloques más pequeños (que forman nuevamente bloques), por ejemplo, de manera iterativa usando partición de árbol cuádruple (QT), partición binaria (BT) o partición de árbol triple (TT) o
 65 cualquier combinación de las mismas, 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 particionado 203 y los modos de predicción se aplican a cada una de las particiones de bloque o subbloques.

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

Particionamiento

- 10 La unidad de particionamiento 262 puede particionar (o dividir) un bloque actual 203 en particiones más pequeñas, por ejemplo, bloques más pequeños de tamaño cuadrado o rectangular. Estos bloques más pequeños (que también pueden denominarse subbloques) pueden dividirse en particiones aún más pequeñas. Esto también se denomina particionamiento de árbol o particionamiento de árbol jerárquico, en donde un
15 bloque raíz, por ejemplo en el nivel de árbol raíz 0 (nivel de jerarquía 0, profundidad 0), se puede particionar recursivamente, por ejemplo, se puede particionar en dos o más bloques de un nivel de árbol inmediatamente inferior, por ejemplo, nodos en el nivel de árbol 1 (nivel de jerarquía 1, profundidad 1), en donde estos bloques se pueden volver a particionar en dos o más bloques de un nivel inmediatamente inferior, por ejemplo, nivel de
20 árbol 2 (nivel de jerarquía 2, profundidad 2), etc., hasta que finalice el particionamiento, por ejemplo, porque se cumple un criterio de terminación, por ejemplo, se alcanza una profundidad de árbol máxima o un tamaño de bloque mínimo. Los bloques que no están particionados en más partes también se denominan bloques hoja o nodos hoja del árbol. Un árbol que utiliza particionamiento en dos se denomina árbol binario (BT), un árbol que utiliza particiones en tres particiones se denomina árbol ternario (TT) y un árbol que utiliza particiones en cuatro particiones se denomina árbol cuádruple (QT).

- 25 Como se mencionó anteriormente, el término "bloque" como se utiliza en esta 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 transformada (TU) y/o a los bloques correspondientes, por ejemplo, un bloque de árbol de codificación (CTB), un bloque de codificación (CB), un bloque de
30 transformada (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 está codificada utilizando tres planos de
35 color separados y estructuras de sintaxis utilizadas 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 modo que la división de un componente en CTB es un particionamiento. 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
40 muestras de una imagen monocromática o una imagen que se codifica utilizando tres planos de color separados y estructuras de sintaxis utilizadas 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 modo que la división de un CTB en bloques de codificación es un particionamiento.

- 45 En ejemplos, por ejemplo, de acuerdo con HEVC, una unidad de árbol de codificación (CTU) se puede dividir en CU mediante el uso de una estructura de árbol cuádruple denominada árbol de codificación. La decisión de codificar un área de imagen usando predicción interimagen (temporal) o intraimagen (espacial) se toma en el nivel de CU. Cada CU se puede dividir 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 en función del tipo de
50 división de PU, una CU se puede particionar en unidades de transformada (TU) según otra estructura de árbol cuádruple similar al árbol de codificación de la CU.

- En ejemplos, por ejemplo, de acuerdo con el último estándar de codificación de vídeo actualmente en
55 desarrollo, al que se hace referencia como Codificación de Vídeo Versátil (VVC), se utiliza la partición 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 hojas cuádruples se particionan además mediante una estructura de árbol binario o de árbol ternario (o triple). Los nodos de hojas del árbol de particionamiento se denominan unidades de codificación (CU), y esa segmentación se utiliza para el procesamiento de predicción y transformada sin ningún particionamiento adicional. 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 propuso usar la partición múltiple, por ejemplo, la partición de árbol triple junto con la estructura de bloques QTBT.

- 60 En un ejemplo, la unidad 260 de selección de modo del codificador de vídeo 20 puede configurarse para realizar

cualquier combinación de las técnicas de particionamiento descritas en esta memoria.

- 5 Como se ha descrito anteriormente, el codificador de vídeo 20 está configurado para determinar o seleccionar el modo de predicción mejor u óptimo de un conjunto de modos de predicción (predeterminados). El conjunto de modos de predicción puede comprender, por ejemplo, modos de intrapredicción y/o modos de interpredicción.

Intrapredicción

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

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

- 20 La unidad de predicción intra 254 (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 de sintaxis 266 para su inclusión en los datos 21 de imagen codificados, de modo que, por ejemplo, el decodificador de vídeo 30 pueda recibir y utilizar los parámetros de predicción para la decodificación.

- 25 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 medio/semipiel y/o de cuarto de pel, o no.

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

- La unidad de interpredicción 244 puede incluir una unidad de estimación de movimiento (ME) y una unidad de compensación de movimiento (MC) (ambas no mostradas en la Figura 2). La unidad de estimación de movimiento está configurada para recibir u obtener el bloque 203 de imágenes (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 en función 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 en función del 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 imágenes. 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.

- 65 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 de vídeo 30 en la decodificación de los bloques

de imágenes del segmento de vídeo.

Codificación de entropía

- 5 La unidad 270 de codificación de entropía está configurada para aplicar, por ejemplo, un algoritmo o esquema de codificación de entropía (por ejemplo, un esquema de codificación de longitud variable (VLC), un esquema de VLC adaptable al contexto (CALVC), 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 (sin compresión) en los
10 coeficientes 209 cuantificados, parámetros de interpredicción, parámetros de intrapredicción, parámetros de filtro de bucle y/u otros elementos de sintaxis para obtener datos de imagen codificados 21 que pueden emitirse a través de la salida 272, por ejemplo, en forma de un flujo de bits codificados 21, de modo que por ejemplo, el decodificador de vídeo 30 pueda recibir y usar los parámetros para decodificar. El flujo 21 de bits codificados
15 puede transmitirse al decodificador 30 de vídeo, o almacenarse en una memoria para su posterior transmisión o recuperación mediante el decodificador de vídeo 30.

- Se pueden usar otras variaciones estructurales del codificador de vídeo 20 para codificar el flujo de vídeo. Por ejemplo, un codificador 20 no basado en transformada puede cuantificar la señal residual directamente sin la
20 unidad 206 de procesamiento de transformada para ciertos bloques o fotogramas. 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

- 25 La Figura 3 muestra un ejemplo de un decodificador de vídeo 30 que está configurado para implementar las técnicas de esta presente invención. El decodificador de vídeo 30 configurado para recibir datos de imágenes codificados 21 (por ejemplo, un flujo de bits codificado 21), por ejemplo, codificados por el codificador 20 para obtener una imagen 331 decodificada. Los datos de imágenes codificados o flujo de bits comprenden
30 información para decodificar los datos de imágenes codificados, por ejemplo, datos que representan bloques de imágenes de un segmento de vídeo codificado y elementos de sintaxis asociados.

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

- 40 Como ya se explicó con respecto al codificador 20, la unidad 210 de cuantificación inversa, la unidad 212 de procesamiento de transformada inversa, la unidad 214 de reconstrucción, el filtro de bucle 220, el búfer de imagen decodificada (DPB) 230, la unidad de interpredicción 344 y la unidad de intrapredicción 354 también se denominan "decodificador incorporado" del codificador de vídeo 20. Por consiguiente, la unidad 310 de
45 cuantificación inversa puede tener una función idéntica a la unidad 110 de cuantificación inversa, la unidad 312 de procesamiento de transformada inversa puede tener una función idéntica a la unidad 212 de procesamiento de transformada inversa, la unidad 314 de reconstrucción puede tener una función idéntica a la unidad 214 de reconstrucción, el filtro de bucle 320 puede tener una función idéntica al filtro de bucle 220, y el búfer de imágenes decodificadas 330 puede tener una función idéntica al búfer de imagen decodificada 230. Por lo
50 tanto, las explicaciones proporcionadas para las respectivas unidades y funciones del codificador de vídeo 20 se aplican correspondientemente a las respectivas unidades y funciones del decodificador de vídeo 30.

Decodificación de entropía

- 55 La unidad 304 de decodificación de entropía está configurada para analizar el flujo de bits 21 (o en general los datos de imágenes codificados 21) y realizar, por ejemplo, decodificación de entropía a los datos de imágenes codificados 21 para obtener, por ejemplo, coeficientes cuantificados 309 y/o parámetros de codificación decodificados (no mostrados en la Figura 3), por ejemplo cualquiera o todos los parámetros de interpredicción (por ejemplo, índice de imágenes de referencia y vector de movimiento), parámetro de intrapredicción (por
60 ejemplo, modo o índice de intrapredicción), parámetros de transformada, 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 puede configurarse además para proporcionar parámetros de
65 interpredicción, parámetros de intrapredicción y/u otros elementos de sintaxis a la unidad 360 de selección de modo y otros parámetros a otras unidades del codificador 30. El decodificador de vídeo 30 puede recibir los

elementos de sintaxis en el nivel de segmento de vídeo y/o el nivel de bloque de vídeo.

Cuantificación inversa

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

15 Transformada inversa

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

Reconstrucción

- 30 La unidad 314 de reconstrucción (por ejemplo, el sumador 314) puede estar configurada para añadir el bloque 313 residual reconstruido 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.

35 Filtrado

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

45 Búfer de imagen decodificada

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

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

55 Predicción

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

Cuando el segmento de vídeo se codifica como un segmento intracodificado (I), la unidad 354 de intrapredicción

de la unidad 360 de modo de selección está configurada para generar el bloque 365 de predicción para un bloque de imagen del segmento de vídeo actual en función de un modo de intrapredicción señalado y datos de bloques previamente decodificados del fotograma o 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 selección de modo está configurada para producir bloques 365 de predicción para un bloque de vídeo del segmento de vídeo actual en función de 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 generarse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de fotogramas de referencia, Lista 0 y Lista 1, usando técnicas de construcción predeterminadas en función de imágenes de referencia almacenadas en DPB 330.

La unidad 360 de selección 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 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 selección de modo usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, 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 actual.

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

Debe entenderse que, en el codificador 20 y el decodificador 30, un resultado de procesamiento de un paso actual puede procesarse aún más y luego enviarse al siguiente paso. Por ejemplo, después del filtrado de interpolación, la derivación del vector de movimiento o el filtrado de bucle, se puede realizar una operación adicional, como Clip o Shift, en el resultado del procesamiento del filtrado de interpolación, la derivación del vector de movimiento o el filtrado de bucle.

Se debe tener en cuenta que se pueden aplicar operaciones adicionales a los vectores de movimiento derivados del bloque actual (incluidos, entre otros, vectores de movimiento de puntos de control de modo afín, vectores de movimiento de subbloques en modos afines, planos, modos ATMVP, vectores de movimiento temporales, etc.). Por ejemplo, el valor del vector de movimiento está restringido a un rango predefinido según el bit que lo representa. Si el bit que representa el vector de movimiento es bitDepth, entonces el rango es $-2^{(\text{bitDepth}-1)} \sim 2^{(\text{bitDepth}-1)}$, donde "A" significa exponenciación. Por ejemplo, si bitDepth se establece en 16, el rango es $-32768 \sim 32767$; si bitDepth se establece en 18, el rango es $-131072 \sim 131071$. Aquí se proporcionan dos métodos para restringir el vector de movimiento.

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

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

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

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

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

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

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

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

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

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

5

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

Método 2: eliminar el MSB de desbordamiento recortando el valor

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

10

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

donde la definición de la función Clip3 es la siguiente:

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

15

La operación "?" se utiliza normalmente como un atajo para representar la condición "si... de lo contrario". Por ejemplo, "X<K?X=1:X=0" se puede interpretar como "si X es menor que K, X se establece igual a 1, de lo contrario (si X no es menor que K) X se establece igual a 0".

20

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

25

El dispositivo 400 de codificación de vídeo comprende puertos 410 de entrada (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 salida (o 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.

30

El procesador 430 está implementado por 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.

40

La memoria 460 puede comprender uno o más discos, unidades de cinta, y unidades de estado sólido y puede ser usada como un dispositivo de almacenamiento de datos en exceso (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 y/o no volátil y puede ser una memoria de solo lectura (ROM), una memoria de acceso aleatorio (RAM), una memoria ternaria de contenido direccionable (TCAM), y/o una memoria estática de acceso aleatorio (SRAM).

50

La Figura 5 es un diagrama de bloques simplificado de un aparato 500 que se puede usar tanto como dispositivo 12 de origen y dispositivo 14 de destino, o como ambos, de la Figura 1 según una realización ejemplar.

5 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.

10 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, incluyendo los programas 510 de aplicación 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.

20 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 funcionar para detectar entradas táctiles. La pantalla 518 se puede acoplar al procesador 502 a través del bus 512.

25 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 varias tarjetas de memoria. Por lo tanto, el aparato 500 puede implementarse en una amplia variedad de configuraciones.

En un ejemplo sobre la construcción de una lista de candidatos a fusión según ITU-T H.265, se construye una lista de candidatos a fusión basada en los siguientes candidatos:

- 35 1. hasta cuatro candidatos espaciales que se deriven de cinco bloques espaciales vecinos,
2. un candidato temporal derivado de dos bloques temporales ubicados conjuntamente, (cubicados),
3. candidatos adicionales, incluidos candidatos bipredictivos combinados y,
- 40 4. Candidatos a vector de movimiento cero.

Candidatos espaciales

45 La información de movimiento de los bloques espaciales vecinos se agrega primero a la lista de candidatos a fusión (en un ejemplo, la lista de candidatos a fusión puede ser una lista vacía antes de que se agregue el primer vector de movimiento a la lista de candidatos a fusión) como candidatos a información de movimiento. Aquí, los bloques vecinos que se consideran insertados en la lista de fusión se ilustran en la Figura 6b. Para la fusión de bloques de interpredicción, se insertan hasta cuatro candidatos en la lista de fusión marcando secuencialmente A1, B1, B0, A0 y B2, en ese orden.

50 La información de movimiento puede contener todos los datos de movimiento, incluida la información sobre si se utilizan una o dos listas de imágenes de referencia, así como un índice de referencia y un vector de movimiento para cada lista de imágenes de referencia.

55 En un ejemplo, después de verificar si un bloque vecino está disponible y contiene información de movimiento, se realizan algunas verificaciones de redundancia adicionales antes de tomar todos los datos de movimiento del bloque vecino como un candidato a información de movimiento. Estas comprobaciones de redundancia se pueden dividir en dos categorías para dos propósitos diferentes:

60 Categoría 1, evitar tener candidatos con datos de movimiento redundantes en la lista,

Categoría 2, evitar la fusión de dos particiones que podrían expresarse por otros medios, lo que crearía una sintaxis redundante.

65 Candidatos temporales

La figura 6a ilustra las coordenadas de los bloques de donde se recuperan los candidatos a información de movimiento temporal. El bloque colocado es el bloque que tiene las mismas coordenadas -x, -y del bloque actual, pero está en una imagen diferente (una de las imágenes de referencia). Los candidatos a información de movimiento temporal se agregan a la lista de fusión si la lista no está completa (en un ejemplo, la lista de fusión no está llena cuando una cantidad de candidatos en la lista de fusión es menor que un umbral, por ejemplo, el umbral puede ser 4, 5, 6, etc.).

Candidatos generados

Después de la inserción de candidatos a información de movimiento espacial y temporal, si la lista de fusión aún no está completa, se agregan candidatos generados para completar la lista. El tamaño de la lista se indica en el conjunto de parámetros de secuencia y es fijo a lo largo de toda la secuencia de vídeo codificada.

Bipredicción

Un modo especial de interpredicción se denomina "bipredicción", donde se utilizan 2 vectores de movimiento para predecir un bloque. Los vectores de movimiento pueden apuntar a imágenes de referencia iguales o diferentes, donde una imagen de referencia puede indicarse mediante un ID de lista de imágenes de referencia y un índice de imágenes de referencia. Por ejemplo, un primer vector de movimiento podría apuntar a una primera imagen en la lista de imágenes de referencia L0 y un segundo vector de movimiento podría apuntar a una primera imagen en la lista de imágenes de referencia L1. Se pueden mantener dos listas de imágenes de referencia (por ejemplo, L0 y L1) y la imagen señalada por el primer vector de movimiento se selecciona de la lista L0 y la imagen señalada por el segundo vector de movimiento se selecciona de la lista L1.

En un ejemplo, si una información de movimiento indica bipredicción, entonces la información de movimiento incluye dos partes:

- Parte L0: Un vector de movimiento y un índice de imágenes de referencia que apunta a una entrada en la lista de imágenes de referencia L0.
- Parte L1: Un vector de movimiento y un índice de imágenes de referencia que apunta a una entrada en la lista de imágenes de referencia L1.

Recuento de orden de imagen (POC): una variable que está asociada con cada imagen, identifica de forma única la imagen asociada entre todas las imágenes en el CVS (secuencia de video codificada) y, cuando la imagen asociada se va a emitir desde el búfer de imágenes decodificadas, indica la posición de la imagen asociada en el orden de salida en relación con las posiciones de orden de salida de las otras imágenes en el mismo CVS que se van a emitir desde el búfer de imágenes decodificadas.

Cada una de las listas de imágenes de referencia L0 y L1 podría incluir una o más imágenes de referencia, cada una de las cuales está identificada con un POC. La asociación con cada índice de referencia y el valor POC podrían señalizarse en el flujo de bits. A modo de ejemplo, las listas de imágenes de referencia L0 y L1 podrían incluir las siguientes imágenes de referencia:

Lista de imágenes de referencia	Índice de referencia	POC
L0	0	12
L0	1	13
L1	0	13
L1	1	14

En el ejemplo anterior, la primera entrada (indicada por el índice de referencia 0) en la lista de imágenes de referencia L1 es la imagen de referencia con valor POC 13. La segunda entrada (indicada por el índice de referencia 1) en la lista de imágenes de referencia L1 es la imagen de referencia con valor POC 14.

El proceso de construcción de la lista de fusión en ITU-T H.265 y en VVC genera una lista de candidatos a información de movimiento. El proceso de construcción de la lista de fusión de VVC se describe en la sección "8.3.2.2 Proceso de derivación de vectores de movimiento de luma para el modo de fusión" del documento JVET-L1001_v2 Codificación de video versátil (borrador 3), que está disponible públicamente en <http://phenix.it-sudparis.eu/jvet/>. El término información de movimiento se refiere a los datos de movimiento que son necesarios para realizar el proceso de predicción de compensación de movimiento. La información de movimiento generalmente se refiere a la siguiente información:

- Si el bloque aplica unipredicción o bipredicción.
 - La ID de la imagen de referencia que se utiliza en la predicción (2 ID si el bloque aplica bipredicción).
- 5 • Vector de movimiento (2 vectores de movimiento si el bloque es bipredicho).
- Información adicional:

10 En VVC y H.265, la lista de candidatos que se genera a partir de la construcción de la lista de fusión incluye información de movimiento de N candidatos. El número N normalmente se incluye en el flujo de bits y puede ser un número entero positivo como 5, 6, etc. Los candidatos que se incluyen en la lista de fusión construida pueden incluir información de unipredicción o información de bipredicción. Esto significa que el candidato seleccionado de la lista de fusión podría indicar una operación de bipredicción.

15 Modo de predicción triangular

20 El concepto del modo de predicción triangular es introducir una nueva partición triangular para la predicción compensada por movimiento. Como se muestra en la Figura 7, se utilizan dos unidades de predicción triangulares para una CU, tanto en dirección diagonal como en dirección diagonal inversa. Cada unidad de predicción triangular para la CU se interpredice utilizando un vector de movimiento de unipredicción y un índice de fotograma de referencia, que se derivan de una lista de candidatos a unipredicción. Se realiza un proceso de ponderación adaptativa en el borde diagonal después de que se hayan predicho las muestras asociadas con cada unidad de predicción triangular, por ejemplo, mediante compensación de movimiento o predicción de intraimagen. Luego, el proceso de transformada y cuantificación se aplica a toda la CU. Se observa que este modo solo se aplica al modo de omisión y al modo de fusión.

30 En el modo de predicción triangular, un bloque se divide en dos partes triangulares (como en la Figura 7), y cada parte se puede predecir utilizando un vector de movimiento. Un vector de movimiento que se utiliza para predecir una parte del triángulo (indicada con PU1) puede ser diferente de un vector de movimiento que se utiliza para predecir la otra parte del triángulo (indicada con PU2). En un ejemplo, se observa que cada parte puede predecirse utilizando solo un único vector de movimiento (unipredicción), a fin de reducir la complejidad de ejecutar el modo de predicción triangular. En otras palabras, la PU1 y la PU2 no se pueden predecir utilizando la bipredicción, que comprende dos vectores de movimiento.

35 Modo de predicción de subbloques

40 El modo de predicción triangular es un caso especial de predicción de subbloques, donde un bloque se divide en dos bloques. En los ejemplos anteriores, se ilustran dos direcciones de división de bloques (particiones de 45 grados y 135 grados). También son posibles otros ángulos de partición y proporciones de partición para la predicción de subbloques (ejemplos en la Figura 8).

En algunos ejemplos, el bloque se divide en 2 subbloques y cada parte (subbloque) se predice con unipredicción.

45 En un ejemplo, de acuerdo con el modo de partición de subbloques utilizado, se aplican los siguientes pasos para obtener muestras de predicción para un bloque:

- Paso 1: dividir un bloque de codificación en 2 subbloques, según un modelo geométrico. Este modelo puede dar como resultado la división del bloque mediante una línea de separación (por ejemplo, una línea recta) como se ejemplifica en las Figura 9 a 12. Se podría entender que el resultado del Paso 1 puede ser sólo un concepto. Por ejemplo, no existe un concepto expreso "subbloque 1" o "subbloque 2" después del Paso 1; pero los parámetros de generación u obtención se refieren a la línea de separación. Las muestras pertenecen a diferentes subbloques según los parámetros que hacen referencia a la línea de separación. En este paso, según un modelo geométrico, las muestras de un bloque de codificación se consideran ubicadas en dos subbloques. El subbloque A o el subbloque B comprende una parte (pero no todas) de las muestras del bloque de codificación actual. El subbloque A o el subbloque B se pueden representar de acuerdo con el signo de una sample_dist de cada muestra. La sample_dist se puede obtener de acuerdo con los ejemplos y realizaciones de los otros párrafos.

- Paso 2: obtener un primer modo de predicción para un primer subbloque y un segundo modo de predicción para un segundo subbloque. En un ejemplo, el primer modo de predicción no es idéntico al segundo modo de predicción. En un ejemplo, un modo de predicción (primer modo de predicción o segundo modo de predicción) puede ser un modo de interpredicción, la información para un modo de interpredicción puede comprender un índice de imágenes de referencia y un vector de movimiento. En otro ejemplo, el modo de predicción puede ser un modo de intrapredicción, la información para un modo de intrapredicción puede

comprender un índice de modo de intrapredicción.

- Paso 3: Obtener los primeros valores de predicción y los segundos valores de predicción, utilizando el primer modo de predicción y el segundo modo de predicción, respectivamente.

- Paso 4: obtener los valores combinados de muestras de predicción de acuerdo con la combinación de los primeros valores de predicción y los segundos valores de predicción, de acuerdo con la división que se describe en el Paso 1. El proceso de combinar muestras de la primera predicción y la segunda predicción para obtener los valores combinados de muestras de predicción de un bloque puede comprender operaciones de filtrado, operaciones de enmascaramiento o copia de muestras.

En un ejemplo, en el paso 1, un bloque de codificación se divide en 2 subbloques de varias maneras. La figura 9 muestra un ejemplo de partición de un bloque de codificación, la línea de separación 1250 divide el bloque en 2 subbloques. Para describir la línea 1250, se señalan dos parámetros, un parámetro es el ángulo α 1210 y el otro parámetro es la distancia d 1230. En algunas realizaciones, el ángulo, como se indica en la Figura 9, se mide entre el eje x y la línea de separación, mientras que la distancia se mide por la longitud del vector que es perpendicular a la línea de separación y que pasa por el centro del bloque actual.

En otro ejemplo, la Figura 10 muestra una forma alternativa de representar la línea de separación, donde los ejemplos del ángulo y la distancia son diferentes con los ejemplos mostrados en la Figura 9.

En algún ejemplo, en el paso 4, se utiliza la división descrita en el paso 1 para la combinación de las dos predicciones mencionadas, para obtener la predicción final. En un ejemplo, se aplica una operación de fusión en el paso 4 para eliminar cualquier artefacto (apariencia arqueada o irregular a lo largo de la línea de separación). La operación de mezcla puede describirse como una operación de filtrado a lo largo de la línea de separación.

En un ejemplo, en el lado del codificador, se determina una línea de separación (los parámetros que definen la línea, por ejemplo, el ángulo y la distancia) en función de una función de costo basada en la tasa de distorsión. Los parámetros de línea determinados se codifican en el flujo de bits. En el lado del decodificador, los parámetros de línea se decodifican (obtienen) de acuerdo con el flujo de bits. En el lado del decodificador, los parámetros de línea se utilizan para la combinación de los primeros valores de predicción y los segundos valores de predicción, para obtener los valores de predicción finales. El paso 1 no requiere que haya dos conceptos de bloque de subcodificación, el decodificador puede usar un bloque de codificación y los parámetros de línea para representar dos conceptos de bloque de subcodificación.

Como hay muchas posibilidades de dividir un bloque de codificación en 2 subbloques, la señalización (codificación) de la división requiere demasiados bits. Y como los valores de ángulo y distancia pueden tener muchos valores diferentes, es necesario señalar demasiada información secundaria en un flujo de bits.

Las realizaciones de la presente invención tratan sobre la señalización de los parámetros de particionamiento. La codificación eficiente se consigue mediante las siguientes características:

- Los parámetros (por ejemplo, ángulo y distancia) para la partición de un bloque se almacenan en una tabla predeterminada, de modo que no es necesario transmitir los valores reales de ángulo y distancia.

- La tabla contiene parámetros de particionamiento que son más probables y que incluyen suficiente variación.

- Un índice de la tabla se codifica en (o se obtiene de) un flujo de bits.

Las realizaciones de la invención logran una manera de codificar conjuntamente los parámetros que describen el particionamiento de un bloque de codificación en subbloques.

Realización 1 (perspectiva del decodificador):

En esta realización, se aplican los siguientes pasos para obtener el valor de predicción combinado de una muestra de un bloque de codificación.

Paso 1: Obtener un valor de un indicador para un bloque de decodificación actual de acuerdo con un flujo de bits.

En una realización, el valor del indicador se utiliza para especificar una forma de particionamiento del modo de fusión de particionamiento geométrico. Por ejemplo, el indicador puede ser `merge_gpm_partition_idx[x0][y0]`, en donde `merge_gpm_partition_idx[x0][y0]` especifica la forma de particionamiento del modo de fusión de

particionamiento geométrico. 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.

- 5 Normalmente, el valor de merge_gpm_partition_idx[x0][y0] se decodifica a partir de un flujo de bits. En un ejemplo, un rango de valores para merge_gpm_partition_idx[][] es de 0 a 63, incluidos 0 y 63. En un ejemplo, un proceso de decodificación para merge_gpm_partition_idx[][] es "bypass".

Cuando merge_gpm_partition_idx[x0][y0] no está presente, se infiere que es igual a 0.

10

Paso 2: Basándose en el valor del indicador y una tabla de búsqueda predefinida para obtener un valor de un primer parámetro y un valor de un segundo parámetro.

(Parameter1, parameter2) = lookupTable (indicator)

15

En un ejemplo, la variable de ángulo de partición angleIdx (parámetro 1) y la variable de distancia distanceIdx (parámetro 2) del modo de particionamiento geométrico se establecen de acuerdo con el valor de merge_gpm_partition_idx[xCb][yCb] (indicador) como se especifica en la siguiente tabla. Se podría entender que, en la implementación, esta relación se puede implementar de acuerdo con la tabla 1 o de acuerdo con una función.

20

Tabla 1: Especificación de angleIdx y distanceIdx según merge_gpm_partition_idx.

merge_gpm_partition_idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
angleIdx	0	0	2	2	2	2	3	3	3	3	4	4	4	4	5	5
distanceIdx	1	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1
merge_gpm_partition_idx	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
angleIdx	5	5	8	8	11	11	11	11	12	12	12	12	13	13	13	13
distanceIdx	2	3	1	3	0	1	2	3	0	1	2	3	0	1	2	3
merge_gpm_partition_idx	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
angleIdx	14	14	14	14	16	16	18	18	18	19	19	19	20	20	20	21
distanceIdx	0	1	2	3	1	3	1	2	3	1	2	3	1	2	3	1
merge_gpm_partition_idx	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
angleIdx	21	21	24	24	27	27	27	28	28	28	29	29	29	30	30	30
distanceIdx	2	3	1	3	1	2	3	1	2	3	1	2	3	1	2	3

- 25 Paso 3: Para una muestra en el bloque de decodificación actual, se calcula una distancia de muestra (sample_dist) de acuerdo con el valor del primer parámetro y el valor del segundo parámetro.

En una realización, el paso 3 comprende:

- 30 Paso 3.1: obtener un valor de índice de un parámetro de ángulo (alphaN o angleIdx) para el bloque actual, un valor de ancho del bloque actual (W), un valor de altura del bloque actual (H). W y H son el ancho y la altura del bloque actual en número de muestras. Por ejemplo, un bloque de codificación con ancho y alto iguales a 8 es un bloque cuadrado que comprende 64 muestras. En otro ejemplo, W y H son el ancho y la altura del bloque actual, en número de muestras de luma. En un ejemplo, el valor del índice del parámetro de ángulo se puede obtener de acuerdo con la descripción anterior haciendo referencia a la tabla 1.

35

Paso 3.2: obtener un valor de una relación whRatio en función del valor de W y el valor de H, representando el valor de whRatio una relación entre el ancho y la altura del bloque de codificación actual.

- 40 En un ejemplo, whRatio = H / W; o whRatio = W/H.

En otro ejemplo, dos variables nCbW y nCbH especifican el ancho y la altura del bloque de codificación actual, y una variable cIdx especifica un índice de componente de color.

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

$nW = (cIdx == 0) ? nCbW : nCbW * SubWidthC;$

$nH = (cIdx == 0) ? nCbH : nCbH * SubHeightC;$

$whRatio = nH / nW.$

- 5 En un ejemplo, las variables SubWidthC y SubHeightC se especifican en la tabla 2, dependiendo de la estructura de muestreo del formato de croma, que se especifica a través de chroma_format_idc (chroma_format_idc especifica el muestreo de croma en relación con el muestreo de luma) y separate_colour_plane_flag (separate_colour_plane_flag igual a 1 especifica que los tres componentes de color del formato de croma 4:4:4 se codifican por separado). separate_colour_plane_flag igual a 0 especifica que los componentes de color no se codifican por separado. Cuando separate_colour_plane_flag no está presente, se infiere que es igual a 0. Cuando separate_colour_plane_flag es igual a 1, la imagen codificada consta de tres componentes separados, cada uno de los cuales consta de muestras codificadas de un plano de color (Y, Cb o Cr) y utiliza la sintaxis de codificación monocromática. En este caso, cada plano de color está asociado con un valor colour_plane_id específico). Es posible que en el futuro ITU-T | ISO/IEC especifique otros valores de chroma_format_idc, SubWidthC y SubHeightC.

Tabla 2: Valores SubWidthC y SubHeightC derivados de chroma_format_idc y separate_colour_plane_flag

chroma_format_idc	separate_colour_plane_flag	Chroma format	SubWidthC	SubHeightC
0	0	Monocromo	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

- 20 En el muestreo monocromático solo hay una matriz de muestra, que nominalmente se considera la matriz de luma.

En el muestreo 4:2:0, cada una de las dos matrices de croma tiene la mitad de la altura y la mitad del ancho de la matriz de luma.

- 25 En el muestreo 4:2:2, cada una de las dos matrices de croma tiene la mitad de la altura y la mitad del ancho de la matriz de luma.

En el muestreo 4:4:4, dependiendo del valor de separate_colour_plane_flag, se aplica lo siguiente:

- 30 - Si separate_colour_plane_flag es igual a 0, cada una de las dos matrices de croma tiene la misma altura y ancho que la matriz de luma.
- De lo contrario (separate_colour_plane_flag es igual a 1), los tres planos de color se procesan por separado como imágenes muestreadas monocromáticas.
- 35

Paso 3.3: Obtener un valor shiftHor de acuerdo con una tabla de búsqueda, el valor de alfa y el valor de whRatio, en un ejemplo, el valor de alfa (alphaN o angleIdx) y el valor de whRatio se utilizan como valores de índice de la tabla de búsqueda. Se podría entender que el valor shiftHor también se puede obtener de acuerdo con una función, en donde el valor de alfa (alphaN o angleIdx) y el valor de whRatio son la entrada de la función y el valor shiftHor es la salida de la función. En un ejemplo, el resultado de la función es similar o igual a la tabla de búsqueda. En un ejemplo, el valor shiftHor representa un tamaño de paso de cuantificación para el proceso de cálculo de la distancia de muestra.

- 45 En otro ejemplo, la función puede representarse por $shiftHor = (angleIdx \% 16 == 8 \parallel (angleIdx \% 16 != 0 \&\& whRatio > 0)) ? 0 : 1.$

Paso 3.4: se calcula un valor de sample_dist según el valor shiftHor.

- 50 En un ejemplo, para calcular la distancia de muestra (sample_dist), primero, las variables offsetX y offsetY se derivan de la siguiente manera: Si shiftHor es igual a 0, se aplica lo siguiente:

$offsetX = (-nW) \gg 1,$

```
offsetY = ( ( -nH ) >> 1 ) +
```

```
( angleIdx < 16 ? ( distanceIdx * nH ) >> 3 : -( ( distanceIdx * nH ) >> 3 ) );
```

De lo contrario (shiftHor es igual a 1), se aplica lo siguiente:

```
offsetX = ( ( -nW ) >> 1 ) +
```

```
( angleIdx < 16 ? ( distanceIdx * nW ) >> 3 : -( ( distanceIdx * nW ) >> 3 ) );
```

```
5 offsetY = ( - nH ) >> 1;
```

- Las variables xL e yL se derivan de la siguiente manera:

```
xL = ( cIdx == 0 ) ? x : x * SubWidthC
```

```
yL = ( cIdx == 0 ) ? y : y * SubHeightC
```

```
sample_dist = ( ( ( xL + offsetX ) << 1 ) + 1 ) * disLut[ displacementX ] +
```

```
(( ( yL + offsetY ) << 1 ) + 1 ) * disLut[ displacementY ].
```

10

Las variables displacementX y displacementY se derivan de la siguiente manera:

```
hwRatio = cbHeight / cbWidth;
```

15

```
displacementX = angleIdx;
```

```
displacementY = ( angleIdx + 8 ) % 32.
```

La matriz disLut se especifica en la tabla 3 de la siguiente manera:

20

Tabla 3 - Especificación de la matriz de distancia de particionamiento geométrico disLut.

idx	0	2	3	4	5	6	8	10	11	12	13	14
disLut[idx]	8	8	8	4	4	2	0	-2	-4	-4	-8	-8
idx	16	18	19	20	21	22	24	26	27	28	29	30
disLut[idx]	-8	-8	-8	-4	-4	-2	0	2	4	4	8	8

25

En otra realización, en un ejemplo, el Paso 3 en las realizaciones anteriores puede comprender los siguientes pasos: Paso 3.1: obtener un valor de un parámetro de ángulo (alphaN o angleIdx) para el bloque actual, un valor de un índice de distancia (distanceIdx), un valor de ancho del bloque actual (W), un valor de altura del bloque actual (H). W y H son el ancho y la altura del bloque actual en número de muestras. Por ejemplo, un bloque de codificación con ancho y alto iguales a 8 es un bloque cuadrado que comprende 64 muestras. En otro ejemplo, W y H son el ancho y la altura del bloque actual, en número de muestras de luma. El valor del índice del parámetro de ángulo se puede obtener de acuerdo con la descripción anterior haciendo referencia a la tabla 1.

30

Paso 3.2: obtener un valor de una relación whRatio de acuerdo con el valor de W y el valor de H, el valor de whRatio representa una relación entre la anchura y la altura del bloque de codificación actual.

35

En un ejemplo, whRatio = H / W; o whRatio = W/H.

En otro ejemplo, dos variables nCbW y nCbH especifican el ancho y la altura del bloque de codificación actual,

y una variable `cldx` especifica un índice de componente de color.

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

5 `nW = (cldx == 0) ? nCbW : nCbW * SubWidthC`

`nH = (cldx == 0) ? nCbH : nCbH * SubHeightC`

`whRatio = nH / nW.`

10

Paso 3.3: Obtener un valor `shiftHor` de acuerdo con una tabla de búsqueda, el valor de `alfa` (`alphaN` o `angleIdx`) y el valor de `whRatio`, en un ejemplo, el valor de `alfa` (`alphaN` o `angleIdx`) y el valor de `whRatio` se utilizan como valores de índice de la tabla de búsqueda. En un ejemplo, el valor `shiftHor` representa un tamaño de paso de cuantificación para el proceso de cálculo de la distancia de muestra. El valor `shiftHor` también se puede obtener de acuerdo con una función, donde el valor de `alfa` (`alphaN` o `angleIdx`) y el valor de `whRatio` son la entrada de la función y el valor `shiftHor` es la salida de la función. En un ejemplo, el resultado de la función es similar o igual a la tabla de búsqueda. En un ejemplo, el valor `shiftHor` representa un tamaño de paso de cuantificación para el proceso de cálculo de la distancia de muestra.

15

20 En otro ejemplo, la función puede representarse por `shiftHor = (angleIdx % 16 == 8 || (angleIdx % 16 != 0 && whRatio > 0)) ? 0 : 1.`

Paso 3.4: se calcula un valor de `sample_dist` según el valor `shiftHor`, el valor de `distanceldx`, el valor del ángulo (`alphaN` o `angleIdx`), el valor de `W` y el valor de `H`.

25

En un ejemplo, para calcular la distancia de muestra (`sample_dist`), primero, las variables `offsetX` y `offsetY` se derivan de la siguiente manera: Si `shiftHor` es igual a 0, se aplica lo siguiente:

`offsetX = (-nW) >> 1,`

30

`offsetY = ((-nH) >> 1) +`

`(angleIdx < 16 ? (distanceldx * nH) >> 3 : -((distanceldx * nH) >> 3));`

De lo contrario (`shiftHor` es igual a 1), se aplica lo siguiente:

`offsetX = ((-nW) >> 1) +`

35

`(angleIdx < 16 ? (distanceldx * nW) >> 3 : -((distanceldx * nW) >> 3));`

`offsetY = (-nH) >> 1;`

- Las variables `xL` e `yL` se derivan de la siguiente manera:

40

`xL = (cldx == 0) ? x : x * SubWidthC`

`yL = (cldx == 0) ? y : y * SubHeightC`

`sample_dist = (((xL + offsetX) << 1) + 1) * disLut[displacementX] +`

`(((yL + offsetY) << 1) + 1) * disLut[displacementY].`

45 Las variables `displacementX` y `displacementY` se derivan de la siguiente manera:

`hwRatio = cbHeight / cbWidth;`

`displacementX = angleIdx;`

`displacementY = (angleIdx + 8) % 32;`

5

La matriz `disLut` se especifica en la tabla 3 anterior.

10 Paso 4: La `sample_dist` calculada se utiliza para calcular factores de ponderación, los factores de ponderación se utilizan para la combinación de un primer valor de predicción y un segundo valor de predicción correspondiente a dicha muestra. En un ejemplo, los factores de ponderación se denominan `sampleWeight1` y `sampleWeight2`, haciendo referencia al peso correspondiente al primer valor de predicción y al peso correspondiente al segundo valor de predicción.

15 En un ejemplo, los factores de ponderación se calculan de acuerdo con las siguientes funciones,

`weightIdxL = partFlip ? 32 + sample_dist : 32 - sample_dist;`

`wValue = Clip3(0, 8, (weightIdxL + 4) >> 3);`

20 En este ejemplo, `wValue` es `sampleWeight1`, y `8-wValue` es `sampleWeight2`. La variable `partFlip` se determina según un valor de `angleIdx`. En un ejemplo, `partFlip = (angleIdx >= 13 && angleIdx <= 27) ? 0 : 1`, o `partFlip = (angleIdx >= 13 && angleIdx <= 27) ? 1 : 0`.

25 Paso 5: El valor combinado de la muestra de predicción en la coordenada de muestra (x,y) se calcula de acuerdo con el primer valor de predicción en la coordenada (x,y), el segundo valor de predicción en la coordenada (x,y), el `sampleWeight1` y el `sampleWeight2`.

En un ejemplo, el valor de muestra de predicción se deriva de la siguiente manera:

`pbSamples[x][y] = Clip3(0, (1 << BitDepth) - 1, (predSamplesLA[x][y] * wValue +`

30 `predSamplesLB[x][y] * (8 - wValue) + offset1) >> shift1);`

35 En donde `bitDepth` representa la profundidad de bits de la muestra, la variable `shift1` se obtiene de acuerdo con `bitDepth`, en un ejemplo, `shift1 = Max(5, 17 - BitDepth)`; la variable `offset1` se obtiene de acuerdo con `shift1`, en un ejemplo, `offset1 = 1 << shift1 - 1`), `predSamplesLA` y `predSamplesLB` son dos matrices $(nCbW) \times (nCbH)$.

Realización 2 (perspectiva del codificador):

En esta realización, se aplican los siguientes pasos.

40

Paso 0: Seleccionar un valor de un primer parámetro y un valor de un segundo parámetro, en un ejemplo, el valor del primer parámetro y el valor del segundo parámetro se obtienen de acuerdo con una métrica de tasa-distorsión.

45 Paso 1: Obtener un valor de índice de acuerdo con el valor del primer parámetro, el valor del segundo parámetro y de acuerdo con una tabla de búsqueda tal que:

`(Parameter1, parameter2) = lookupTable(index)`

50 Paso 2: Codificar el valor del índice en un flujo de bits.

Opcionalmente, en esta realización se utilizan los siguientes pasos,

55 Paso 3: Para una muestra en un bloque de decodificación, se calcula una distancia de muestra (`sample_dist`) de acuerdo con el valor del primer parámetro y el valor del segundo parámetro.

Paso 4: La `sample_dist` calculada se utiliza para calcular factores de ponderación para la combinación de un primer valor de predicción y un segundo valor de predicción correspondiente a dicha muestra. Los factores de ponderación se denominan `sampleWeight1` y `sampleWeight2`, haciendo referencia al peso correspondiente al

primer valor de predicción y al peso correspondiente al segundo valor de predicción.

Paso 5: El valor de predicción combinado para la muestra en la coordenada (x,y) se calcula de acuerdo con el primer valor de predicción en la coordenada (x,y), el segundo valor de predicción en la coordenada (x,y), sampleWeight1 y sampleWeight2.

Según un ejemplo, la tabla de búsqueda en la realización 1 es la misma que la tabla de búsqueda en la realización 2, de modo que el codificador y el decodificador pueden obtener el mismo resultado.

En algunos ejemplos, la distancia de muestra puede representar la distancia horizontal o la distancia vertical, o una combinación de distancia vertical y horizontal, de dicha muestra a una línea de separación (la línea de separación se utiliza para dividir un bloque de codificación en dos subbloques). Dicha muestra está representada por unas coordenadas (x,y) con respecto a la muestra superior izquierda del bloque de codificación. La coordenada de muestra y la distancia de muestra (sample_dist) se ejemplifican en la Figura 11 y en la Figura 12.

En un ejemplo, el primer parámetro representa un valor de ángulo cuantificado (angleIdx) y el segundo parámetro representa un valor de distancia cuantificado (distanceldx). Los dos parámetros describen una ecuación lineal. En un ejemplo, la distancia 1230 se puede obtener de acuerdo con distanceldx (segundo parámetro), y el ángulo alfa (1210) se puede obtener de acuerdo con angleIdx (primer parámetro). La distancia 1230 puede ser una distancia al centro de un bloque de codificación, y el ángulo puede ser un ángulo entre la línea de separación y la línea horizontal (o equivalentemente la vertical) que pasa por el punto central del bloque de codificación.

Según la invención, la tabla de búsqueda está predefinida.

En un ejemplo, una distancia de muestra sample_dist se calcula de acuerdo con la fórmula: $\text{sample_dist} = ((x < 1) + 1) * \text{Dis}[\text{angleIdx1}] + ((y < 1) + 1) * \text{Dis}[\text{angleIdx2}] - \text{offset}(\text{distanceldx})$. Donde,

- El valor de angleIdx1 y el valor de angleIdx2 son del flujo de datos (bitstream) o se derivan/calculan en base a otra información obtenida del flujo de datos (bitstream), angleIdx1 y angleIdx2 representan parámetros trigonométricos cuantificados de una línea de separación, el primero de tipo coseno y el segundo de tipo seno. En un ejemplo, x e y son las coordenadas -x e -y de una muestra con respecto a la muestra superior izquierda de un bloque de codificación.

- Dis[] es una tabla de búsqueda. Dis[angleIdx1] describe el cambio en la distancia de muestra (sample_dist) con respecto a un aumento de unidad (un aumento de valor 1) en la coordenada -x de dicha muestra. Dis[angleIdx2] describe el cambio en la distancia de muestra (sample_dist) con respecto a un aumento de unidad (un aumento de valor 1) en la coordenada -y de dicha muestra. En un ejemplo, angleIdx1 es igual a angleIdx y angleIdx2 es igual a (displacementX + 8)%32.

- offset(distanceldx) es un valor de desplazamiento que es una función de un valor de índice (distanceldx), el valor de índice se obtiene del flujo de datos o se deriva/calcula en función de otra información obtenida del flujo de datos.

Construcción de la tabla de búsqueda:

De acuerdo con la invención, un rango de valores para el primer parámetro puede estar comprendido entre 0 y $4K-1$, donde K es un número entero mayor que 0; un rango de valores para el segundo parámetro puede estar comprendido entre 0 y N. En este caso, las filas de dicha tabla de búsqueda pueden incluir todas las combinaciones del primer parámetro y del segundo parámetro excepto las combinaciones:

- Primer conjunto de pares: un valor del segundo parámetro es igual a 0 y un valor del primer parámetro es igual a uno de {0, K, 2K, 3K}.

- Segundo conjunto de pares: un valor del segundo parámetro = 0, un valor del primer parámetro igual a uno de {K/2, 3K/2, 5K/2, 7K/2}.

- Tercer conjunto de parámetros: un valor del segundo parámetro = 0, un valor del primer parámetro mayor o igual a 2K.

Un valor de K es 8 o 4. El número K describe cuántos valores de ángulo discretos se utilizan para dividir 90 grados de ángulo. Si el número K es 8, la línea de separación podría tener los siguientes valores de ángulo: 0, 11,25, 22,5, 33,75, 45, ... grados. Si el valor de K es 4, entonces los valores de los ángulos son: 0, 22,5, 45, 67,5, ... grados.

En un ejemplo, un valor de N es 4. En general, N puede ser un número entero mayor que 0.

5 El segundo parámetro describe una distancia de la línea de separación al centro de un bloque de codificación. Si un valor del segundo parámetro es cero, esto significa que la línea de separación pasa por el centro del bloque de codificación. Si un valor del primer parámetro es igual a K, esto significa que la línea de separación está orientada horizontal o verticalmente (el valor del ángulo es igual a 0, 90, 270 o 360 grados). El primer conjunto de pares se excluye de la tabla de búsqueda ya que esto crearía una partición redundante, que también se puede lograr mediante el particionamiento binario (es decir, en dos) de un bloque de codificación con un árbol binario. La división binaria es un mecanismo que particiona un bloque de codificación en dos bloques de codificación de igual tamaño a lo largo del eje vertical u horizontal (y no en subbloques). Por lo tanto, en la presente invención, el primer conjunto de pares se excluye de la tabla de búsqueda.

15 El segundo conjunto de pares se excluye de la tabla de búsqueda, ya que el segundo conjunto de pares crearía divisiones que se pueden lograr mediante el modo de predicción triangular. Por lo tanto, excluir el segundo conjunto de pares evitaría particiones de subbloques redundantes.

20 El tercer conjunto de pares se excluye de la tabla ya que cuando el ángulo es igual a X o X+180, se crearían particiones idénticas si el segundo parámetro es igual a cero.

Según un ejemplo, los siguientes pares pueden excluirse de la tabla de búsqueda:

- Un valor del segundo parámetro igual a N y un valor del primer parámetro igual a uno de {K/2, 3K/2, 5K/2, 7K/2}.
- Un valor del segundo parámetro igual a N y un valor del primer parámetro igual a uno de {K/2-1, 3K/2-1, 5K/2-1, 7K/2-1, K/2+1, 3K/2+1, 5K/2+1, 7K/2+1}.

30 Los pares anteriores podrían excluirse de la tabla de búsqueda ya que crean 2 subbloques, uno de los cuales podría ser demasiado pequeño. Si un valor del primer parámetro es igual a K/2, 3K/2, ..., esto significa que el ángulo de la línea de separación es de 45 grados, 135 grados, 225 grados o 315 grados. Si un valor del segundo parámetro es igual a N (que es el valor más alto), entonces uno de los subbloques comprendería una pequeña parte de una esquina del bloque de codificación, que sería demasiado pequeña para la operación de división.

Ordenación de pares en la tabla de búsqueda:

40 Según un ejemplo, los pares (primer parámetro, segundo parámetro) se pueden agregar al final de la tabla de búsqueda, si el segundo parámetro es igual a N (donde N es el valor más alto posible del segundo parámetro). En otras palabras, si hay P pares donde el valor del segundo parámetro es igual a N, entonces las últimas P entradas de la tabla de búsqueda pueden comprender dicho par P. Por ejemplo, si N es igual a 4 y K es igual a 8, entonces hay $P = 4K*1 = 32$ pares de parámetros para los cuales el segundo parámetro es igual a 4.

45 Según otro ejemplo, los pares de parámetros se ordenan según un proceso heurístico. Este proceso heurístico puede determinarse a partir de datos de entrenamiento que, por ejemplo, reflejan la aparición o el uso de pares de parámetros en el conjunto de datos de entrenamiento. De este modo, los pares de parámetros que aparecen con mayor frecuencia se colocan al principio de la lista, mientras que los pares de parámetros que aparecen con menor frecuencia se colocan al final de la lista.

50 Cuando el segundo parámetro es igual a N, un subbloque es más pequeño que el otro subbloque. Es menos probable seleccionar dicha partición (en comparación con los casos en los que el número de muestras en cada subbloque es más similar). Además, un valor de índice (de la tabla de búsqueda) que tiene un valor más alto generalmente requiere una mayor cantidad de bits para ser codificado en un flujo de bits. Por lo tanto, aumenta la eficiencia de la compresión al incluir pares menos probables al final de la tabla, por lo que el comienzo de la tabla se puede asignar a pares más probables (por ejemplo, pares que resultan en particiones de subbloques más favorables).

60 A continuación se muestra un ejemplo del indicador (geo_partition_idx) y la tabla de búsqueda. El indicador tiene un valor entre 0 y 139, que se utiliza para seleccionar el primer parámetro y el segundo parámetro mediante una tabla de búsqueda. Un ejemplo de la tabla de búsqueda es la Tabla4.

Tabla 4: tabla de búsqueda donde geo_partition_idx es el índice de la tabla, y angleidx es el primer parámetro y distanceldx es el segundo parámetro.

ES 3 024 974 T3

geo_partition_idx	angleIdx	distanceIdx
0	0	1
1	0	2
2	0	3
3	0	4
4	1	0
5	1	1
6	1	2
7	1	3
8	1	4
9	2	0
10	2	1
11	2	2
12	2	3
13	2	4
14	3	0
15	3	1
16	3	2
17	3	3
18	3	4
19	4	1
20	4	2
21	4	3
22	4	4
23	5	0
24	5	1
25	5	2
26	5	3
27	5	4
28	6	0
29	6	1
30	6	2
31	6	3
32	6	4
33	7	0
34	7	1
35	7	2
36	7	3
37	7	4
38	8	1
39	8	2
40	8	3
41	8	4
42	9	0
43	9	1
44	9	2
45	9	3
46	9	4
47	10	0
48	10	1
49	10	2

ES 3 024 974 T3

geo_partition_idx	angleidx	distanceidx
50	10	3
51	10	4
52	11	0
53	11	1
54	11	2
55	11	3
56	11	4
57	12	1
58	12	2
59	12	3
60	12	4
61	13	0
62	13	1
63	13	2
64	13	3
65	13	4
66	14	0
67	14	1
68	14	2
69	14	3
70	14	4
71	15	0
72	15	1
73	15	2
74	15	3
75	15	4
76	16	1
77	16	2
78	16	3
79	16	4
80	17	1
81	17	2
82	17	3
83	17	4
84	18	1
85	18	2
86	18	3
87	18	4
88	19	1
89	19	2
90	19	3
91	19	4
92	20	1
93	20	2
94	20	3
95	20	4
96	21	1
97	21	2
98	21	3
99	21	4

geo_partition_idx	angleidx	distanceidx
100	22	1
101	22	2
102	22	3
103	22	4
104	23	1
105	23	2
106	23	3
107	23	4
108	24	1
109	24	2
110	24	3
111	24	4
112	25	1
113	25	2
114	25	3
115	25	4
116	26	1
117	26	2
118	26	3
119	26	4
120	27	1
121	27	2
122	27	3
123	27	4
124	28	1
125	28	2
126	28	3
127	28	4
128	29	1
129	29	2
130	29	3
131	29	4
132	30	1
133	30	2
134	30	3
135	30	4
136	31	1
137	31	2
138	31	3
139	31	4

Se podría entender que la tabla 4 es solo un ejemplo para mostrar posibles valores de combinación para el primer parámetro y el segundo parámetro, en otras implementaciones, parte de la tabla 4 se puede utilizar como tabla de búsqueda para obtener el primer parámetro y el segundo parámetro.

5

Ejemplo 1. Un método de codificación implementado por un dispositivo de decodificación, que comprende:

obtener un flujo de bits;

10 obtener un valor de un indicador para un bloque actual de acuerdo con el flujo de bits.

obtener un valor de un primer parámetro para el bloque actual y un valor de un segundo parámetro para el bloque actual, de acuerdo con el valor del indicador y una tabla de búsqueda predefinida.

obtener un valor de una distancia de muestra para una muestra que se encuentra en el bloque actual, de acuerdo con el valor del primer parámetro y el valor del segundo parámetro.

- 5 obtener un valor de predicción para la muestra, de acuerdo con el valor de la distancia de muestra para la muestra.

Ejemplo 2. El método del ejemplo 1, en donde el primer parámetro representa un ángulo para particionar el bloque actual.

10

Ejemplo 3. El método del ejemplo 1 ó 2, en donde el segundo parámetro representa una distancia para particionar el bloque actual.

- 15 Ejemplo 4. El método de una cualquiera de los ejemplos 1 a 3, en donde la tabla de búsqueda predefinida está compuesta por pares de parámetros primero y segundo (en un ejemplo, cada entrada en la tabla es un par único de primer parámetro y segundo parámetro), el siguiente par está en la tabla de búsqueda predefinida; el primer parámetro representa un ángulo correspondiente a uno de los ángulos de 0, 45, 90, 135, 180, 225, 270 o 315 grados, y el segundo parámetro correspondiente del par no representa una distancia de cero muestras desde el centro del bloque actual.

20

Ejemplo 5. El método de una cualquiera de los ejemplos 1 a 4, en donde la tabla de búsqueda predefinida está compuesta por pares de parámetros primero y segundo (en un ejemplo, cada entrada en la tabla es un par único de primer parámetro y segundo parámetro), el siguiente par está en la tabla de búsqueda predefinida; el primer parámetro representa un ángulo no correspondiente a uno de los ángulos de 0, 45, 90, 135, 180, 225, 270 o 315 grados, y el segundo parámetro correspondiente del par representa una distancia de cero muestras desde el centro del bloque actual.

25

Ejemplo 6. El método de una cualquiera de los ejemplos 1 a 5, en donde la tabla de búsqueda predefinida está compuesta por pares de parámetros primero y segundo (en un ejemplo, cada entrada en la tabla es un par único de primer parámetro y segundo parámetro), el siguiente par no está comprendido en la tabla de búsqueda predefinida; el primer parámetro representa un ángulo no correspondiente a uno de los ángulos de 0, 45, 90, 135, 180, 225, 270 o 315 grados, y el segundo parámetro correspondiente del par representa una distancia de cero muestras desde el centro del bloque actual.

30

- 35 Ejemplo 7: Un decodificador (30) que comprende circuitos de procesamiento para llevar a cabo el método según una cualquiera de los ejemplos 1 a 6.

Ejemplo 8: Un producto de programa informático que comprende un código de programa para realizar el método según una cualquiera de los ejemplos 1 a 6.

40

Ejemplo 9. Un decodificador, que comprende:

uno o más procesadores; y

- 45 un medio 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 para llevar a cabo el método de acuerdo con uno cualquiera de los ejemplo 1 a 6.

- 50 A continuación se presenta 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 los utiliza.

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

60

El dispositivo de captura 3102 genera datos y puede codificar los datos mediante el método de codificación como se muestra en las realizaciones anteriores. Alternativamente, el dispositivo de captura 3102 puede distribuir los datos a un servidor de transmisión (no mostrado en la

65

Figuras), y el servidor codifica los datos y transmite los datos codificados al dispositivo terminal 3106. El

dispositivo de captura 3102 incluye, entre otros, una cámara, un teléfono inteligente o Pad, un ordenador o portátil, un sistema de videoconferencia, una PDA, un dispositivo montado en un vehículo o una combinación de cualquiera de ellos, o similares. Por ejemplo, el dispositivo de captura 3102 puede incluir el dispositivo fuente 12 como se describió anteriormente. Cuando los datos incluyen vídeo, el codificador de vídeo 20 incluido en el dispositivo de captura 3102 puede realmente realizar el procesamiento de codificación de vídeo. Cuando los datos incluyen audio (es decir, voz), un codificador de audio incluido en el dispositivo de captura 3102 puede realmente realizar el procesamiento de codificación de audio. Para algunos escenarios prácticos, el dispositivo de captura 3102 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 de captura 3102 distribuye los datos de audio codificados y los datos de vídeo codificados al dispositivo terminal 3106 por separado.

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

Para un dispositivo terminal con su pantalla, por ejemplo, un teléfono inteligente o Pad 3108, un ordenador o portátil 3110, una grabadora de vídeo en red (NVR)/grabadora de vídeo digital (DVR) 3112, una TV 3114, un asistente digital personal (PDA) 3122 o un dispositivo montado en un vehículo 3124, el dispositivo terminal puede alimentar los datos decodificados a su pantalla. Para un dispositivo terminal no equipado con pantalla, por ejemplo, STB 3116, sistema de videoconferencia 3118 o sistema de videovigilancia 3120, se contacta con una pantalla externa 3126 para recibir y mostrar los datos decodificados.

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

La Figura 14 es un diagrama que muestra una estructura de un ejemplo del dispositivo terminal 3106. Después de que el dispositivo terminal 3106 recibe la transmisión del dispositivo de captura 3102, la unidad de procesamiento de protocolo 3202 analiza el protocolo de transmisión del flujo. El protocolo incluye, entre otros, el Protocolo de transmisión en tiempo real (RTSP), el Protocolo de transferencia de hipertexto (HTTP), el Protocolo de transmisión en vivo HTTP (HLS), MPEG-DASH, el Protocolo de transporte en tiempo real (RTP), el Protocolo de mensajería en tiempo real (RTMP) o cualquier tipo de combinación de los mismos o similares.

Después de que la unidad 3202 de procedimiento de protocolo procesa la secuencia, se genera un archivo de secuencia. El archivo se envía a una unidad demultiplexora 3204. La unidad demultiplexora 3204 puede separar los datos multiplexados en los datos de audio codificados y los datos de vídeo codificados. Como ya se describió anteriormente, para algunos 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 de vídeo 3206 y al decodificador de audio 3208 sin pasar por la unidad demultiplexora 3204.

A través del procesamiento de demultiplexación, se generan flujos elementales de vídeo (ES), ES de audio y, opcionalmente, subtítulos. El decodificador de vídeo 3206, que incluye el decodificador de vídeo 30 como se explicó en las realizaciones mencionadas anteriormente, decodifica el vídeo ES mediante el método de decodificación como se muestra en las realizaciones mencionadas anteriormente para generar un fotograma de vídeo, y envía estos datos a la unidad síncrona 3212. El decodificador de audio 3208 decodifica el audio ES para generar una secuencia de audio y envía estos datos a la unidad síncrona 3212. Como alternativa, el fotograma de vídeo puede almacenarse en un búfer (no mostrado en la Figura 14) antes de enviarlo a la unidad síncrona 3212. Como alternativa, la secuencia de audio puede almacenarse en un búfer (no mostrado en la Figura 14) antes de enviarlo a la unidad síncrona 3212.

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

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

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

Operadores matemáticos

- 10 Los operadores matemáticos utilizados en esta invención son similares a los utilizados en el lenguaje de programación C. Sin embargo, los resultados de la división de números enteros y de las operaciones de desplazamiento aritmético se definen con mayor precisión y se definen operaciones adicionales, 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

- 20 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 y. En otros contextos, dicha notación se utiliza para superíndices y no para interpretación 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. Se utiliza para indicar división en ecuaciones matemáticas donde no se pretende truncamiento ni redondeo.
$\frac{x}{y}$	Se utiliza para indicar división en ecuaciones matemáticas donde no se pretende truncamiento ni redondeo.
$\sum_{i=x}^y f(i)$	La suma de f(i) donde i toma todos los valores enteros desde x hasta y inclusive.
$x \% y$	Módulo. Resto de x dividido por y, definido sólo para números enteros x e y con $x \geq 0$ e $y > 0$.

Operadores lógicos

- 25 Los siguientes operadores lógicos se definen de la siguiente manera:

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

Operadores relacionales

- 30 Los siguientes operadores relacionales se definen de la siguiente manera:

>	Mayor que
>=	Mayor que o igual a
<	Menor que
<=	Menor que o igual a
=	Igual a
!=	No es igual a

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. El valor

"na" no se considera igual a ningún otro valor.

Operadores bit a bit

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

- & "Y" bit a bit. Al operar con argumentos enteros, se opera sobre una representación de complemento a dos del valor entero. Cuando se opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se extiende agregando más bits significativos iguales a 0.
- | "O" bit a bit. Al operar con argumentos enteros, se opera sobre una representación de complemento a dos del valor entero. Cuando se opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se extiende agregando más bits significativos iguales a 0.
- ^ "O exclusivo" bit a bit. Al operar con argumentos enteros, se opera sobre una representación de complemento a dos del valor entero. Cuando se opera con un argumento binario que contiene menos bits que otro argumento, el argumento más corto se extiende agregando más bits significativos iguales a 0.
- x >> y Desplazamiento aritmético a la derecha de una representación entera en complemento a dos de x por y dígitos binarios. Esta función se define solo para valores 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 dígitos binarios x por y. Esta función se define solo para valores 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

- 10 Los siguientes operadores aritméticos se definen de la siguiente manera:

Operadores 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.
- Disminución, 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 disminución.
- += Incrementar por la cantidad especificada, es decir, x += 3 es equivalente a x = x + 3, y x += (-3) es equivalente a x = x + (-3).
- = Disminuir por la cantidad especificada, es decir, x -= 3 es equivalente a x = x - 3, y x -= (-3) es equivalente a x = x - (-3).

Nota de rango

- 15 La siguiente nota se utiliza para especificar un rango de valores:

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

Funciones matemáticas

- 20 Se definen las siguientes funciones matemáticas:

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

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

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

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

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

5 $\text{Clip1r}(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}) - 1, x)$

$$\text{Clip1c}(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}) - 1, x)$$

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

10

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

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

15

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

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

20

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

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

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

25

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

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

30

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

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

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

35

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

Orden de precedencia de operaciones

40

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.

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

5

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

10

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

Descripción textual de operaciones lógicas

15

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

si(condición 0)

afirmación 0

20

si no si(condición 1)

afirmación 1

25

...

Si no /* observación informativa sobre la condición actual */

Afirmación n

30

puede describirse de la siguiente manera:

... de la siguiente manera / ... se aplica lo siguiente:

35

- Si condición 0, afirmación 0

- De lo contrario, si la condición 1, afirmación 1 - ...

- En caso contrario (observación informativa sobre el estado restante), afirmación n.

40

Cada "Si... De lo contrario, si... De lo contrario, la afirmación ..." 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 afirmaciones se pueden identificar haciendo coincidir "... como sigue" o "... se aplica lo siguiente" con la terminación "De lo contrario, ...". En el texto, un planteamiento de operaciones lógicas como se describiría matemáticamente en la siguiente forma:

si(condición 0a && condición 0b)

10 planteamiento 0

si no si(condición 1a || condición 1b)

15 planteamiento 1

...

si no el

20 planteamiento n

se puede describir de la siguiente manera:

... de la siguiente manera / ... se aplica lo siguiente:

25

- Si todas las condiciones siguientes son verdaderas, afirmación 0:

- condición 0a

30

- condición 0b

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

- condición 1a

35

- condición 1b

- En caso contrario, la afirmación n

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

si(condición 0)

afirmación 0

45

si(condición 1)

afirmación 1

50 se puede describir de la siguiente manera:

Cuando condición 0, afirmación 0

Cuando condición 1, afirmación 1.

55

Aunque las realizaciones de la invención se han descrito principalmente en base a la codificación de vídeo, debe tenerse en cuenta 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 aquí también pueden configurarse para el procesamiento o codificación de imágenes fijas, es decir, el procesamiento o codificación de una imagen individual independientemente de cualquier imagen precedente o consecutiva como en la codificación de vídeo. En general, es posible que solo las unidades de interpredicción 244 (codificador) y 344 (decodificador) no estén disponibles en caso de que la codificación del procesamiento de imágenes esté limitada a una sola imagen 17. Todas las demás funcionalidades (también denominadas herramientas o tecnologías) del codificador de vídeo 20 y del decodificador de vídeo 30 pueden utilizarse igualmente para el procesamiento de imágenes fijas, por ejemplo, cálculo residual 204/304, transformada 206, cuantificación 208, cuantificación inversa 210/310, transformada (inversa) 212/312, particionamiento 262/362, intrapredicción 254/354 y/o filtrado de bucle 220,

65

320 y codificación de entropía 270 y decodificación de entropía 304. Las realizaciones, por ejemplo del codificador 20 y del decodificador 30, y las funciones descritas en esta memoria, por ejemplo con referencia al codificador 20 y al decodificador 30, pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se utiliza software para la implementación, las funciones pueden almacenarse en un medio legible por ordenador o transmitirse a través de medios de comunicación como una o más instrucciones o códigos y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como un medio de almacenamiento de datos, o medios de comunicación incluyendo cualquier medio que facilite la transferencia de un programa de ordenador de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, el medio legible por ordenador puede corresponder generalmente a: (1) un medio de almacenamiento legible por ordenador tangible que es no transitorio o (2) un medio de comunicaciones tal como una señal o una onda portadora. El medio de almacenamiento de datos puede ser cualquier medio utilizable al que puedan acceder uno o más ordenadores o uno o más procesadores para recuperar instrucciones, códigos y/o estructuras de datos para implementar las tecnologías descritas en esta invención. Un producto de programa informático puede incluir el medio legible por ordenador.

A modo de ejemplo pero no limitación, tal soporte de almacenamiento legible por ordenador puede incluir una RAM, una ROM, una EEPROM, un CD-ROM u otro aparato de almacenamiento de disco óptico, un aparato de almacenamiento de disco magnético u otro aparato de almacenamiento magnético, una memoria flash, o cualquier otro medio que se pueda configurar para almacenar código de programa deseado en forma de una instrucción o una estructura de datos y al que pueda acceder un ordenador. Además, cualquier conexión se puede denominar correctamente medio legible por ordenador. Por ejemplo, si se transmiten instrucciones desde un sitio web, un servidor u otra fuente remota mediante un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, el DSL o las tecnologías inalámbricas como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que el medio de almacenamiento legible por ordenador y el medio de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que en realidad significan soportes de almacenamiento tangibles no transitorios. El disco magnético y el disco usados en esta 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 magnéticos generalmente reproducen datos magnéticamente, mientras que los discos reproducen datos ópticamente mediante láser. También deben incluirse combinaciones de los antemencionados dentro del alcance de los medios 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 (ASICs), matrices de puerta programables de campo (FPGAs), u otros circuitos integrados equivalentes o circuitos lógicos discretos. En consecuencia, el término "procesador", tal como se utiliza en esta memoria, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en esta memoria. Además, en algunos aspectos, la funcionalidad descrita en esta memoria puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificación y decodificación, o incorporados en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta invención pueden implementarse en una amplia variedad de dispositivos o aparatos, incluidos un teléfono inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta invención para enfatizar aspectos funcionales de los aparatos configurados para realizar las técnicas descritas, pero no se implementan necesariamente mediante el uso de diferentes unidades de hardware. En lugar de ello, como se describió anteriormente, varias unidades pueden combinarse en una unidad de hardware de códec o proporcionarse mediante una colección de unidades de hardware interoperativas, incluidos uno o más procesadores como se describió anteriormente, junto con un software y/o firmware adecuado.

REIVINDICACIONES

1. Un método de codificación implementado por un dispositivo de decodificación, que comprende:
 - 5 obtener de un flujo de bits;
obtener un valor de índice de una tabla de búsqueda predefinida para un bloque actual a partir del flujo de bits;
obtener un valor de un primer parámetro para el bloque actual y un valor de un segundo parámetro para el
10 bloque actual en función del valor del índice de la tabla de búsqueda predefinida de la siguiente manera:
(primer parámetro, segundo parámetro) = lookupTable(valor de índice),
en donde el primer y el segundo parámetro describen una línea de separación que divide el bloque actual en
15 dos subbloques, representando el primer parámetro un ángulo para dividir el bloque actual y representando el
segundo parámetro una distancia para particionar el bloque actual, en donde un rango de valores para el primer
parámetro está entre 0 y $4K-1$, y un rango de valores para el segundo parámetro está entre 0 y N, donde K es
igual a 4 u 8, y N es un número entero mayor que 0, y en donde la tabla de búsqueda incluye todas las
20 combinaciones del primer parámetro y del segundo parámetro excepto las siguientes combinaciones:
- un valor del segundo parámetro es igual a 0, y un valor del primer parámetro es igual a uno de {0, K, 2K, 3K},
- un valor del segundo parámetro igual a 0 y un valor del primer parámetro igual a uno de {K/2, 3K/2, 5K/2,
25 7K/2}, y
- un valor del segundo parámetro es igual a 0 y un valor del primer parámetro es mayor que 2K;
obtener un valor de una distancia de muestra para una muestra que se encuentra en el bloque actual, de
30 acuerdo con el valor del primer parámetro y el valor del segundo parámetro.
obtener un valor de predicción para la muestra, de acuerdo con el valor de la distancia de muestra para la
muestra.
 2. El método de la reivindicación 1, en donde la obtención de un valor de predicción para la muestra de acuerdo
35 con el valor de la distancia de muestra para la muestra comprende:
calcular dos factores de ponderación en función del valor de la distancia de la muestra; y
obtener el valor de predicción para la muestra de acuerdo con un primer valor de predicción, un segundo valor
40 de predicción y los dos factores de ponderación.
 3. El método de cualquiera de las reivindicaciones 1 a 2, en donde el valor de la distancia de la muestra
representa la distancia horizontal o la distancia vertical, o una combinación de las distancias vertical y
45 horizontal, de dicha muestra hasta la línea de separación.
 4. Un decodificador, que comprende:
uno o más procesadores; y
50 un medio de almacenamiento acoplado a uno o más procesadores y que almacena la programación para su
ejecución por los procesadores, en donde la programación, cuando la ejecutan los procesadores, configura el
decodificador para llevar a cabo el método según las reivindicaciones 1 a 3.
 5. Un método de codificación implementado por un dispositivo de codificación, que comprende:
55 seleccionar un valor de un primer parámetro y un valor de un segundo parámetro para un bloque actual, en
donde el primer y el segundo parámetro describen una línea de separación que divide el bloque actual en dos
subbloques, con el primer parámetro representando un ángulo para particionar el bloque actual, y el segundo
parámetro representando una distancia para particionar el bloque actual, en donde un rango de valores para
60 el primer parámetro está entre 0 y $4K-1$, y un rango de valores para el segundo parámetro está entre 0 y N,
donde K es igual a 4 u 8, y N es un número entero mayor que 0;
obtener un valor de índice de una tabla de búsqueda predefinida para el bloque actual en función del valor del
primer parámetro y el valor del segundo parámetro de la siguiente manera:
65 (primer parámetro, segundo parámetro) = lookupTable(valor de índice),

donde la tabla de búsqueda incluye todas las combinaciones del primer parámetro y del segundo parámetro excepto las siguientes combinaciones:

- 5 - un valor del segundo parámetro es igual a 0, y un valor del primer parámetro es igual a uno de {0, K, 2K, 3K},
- un valor del segundo parámetro igual a 0 y un valor del primer parámetro igual a uno de {K/2, 3K/2, 5K/2, 7K/2}, y
- 10 - un valor del segundo parámetro es igual a 0 y un valor del primer parámetro es mayor que 2K;
codificar el valor del índice en un flujo de bits.

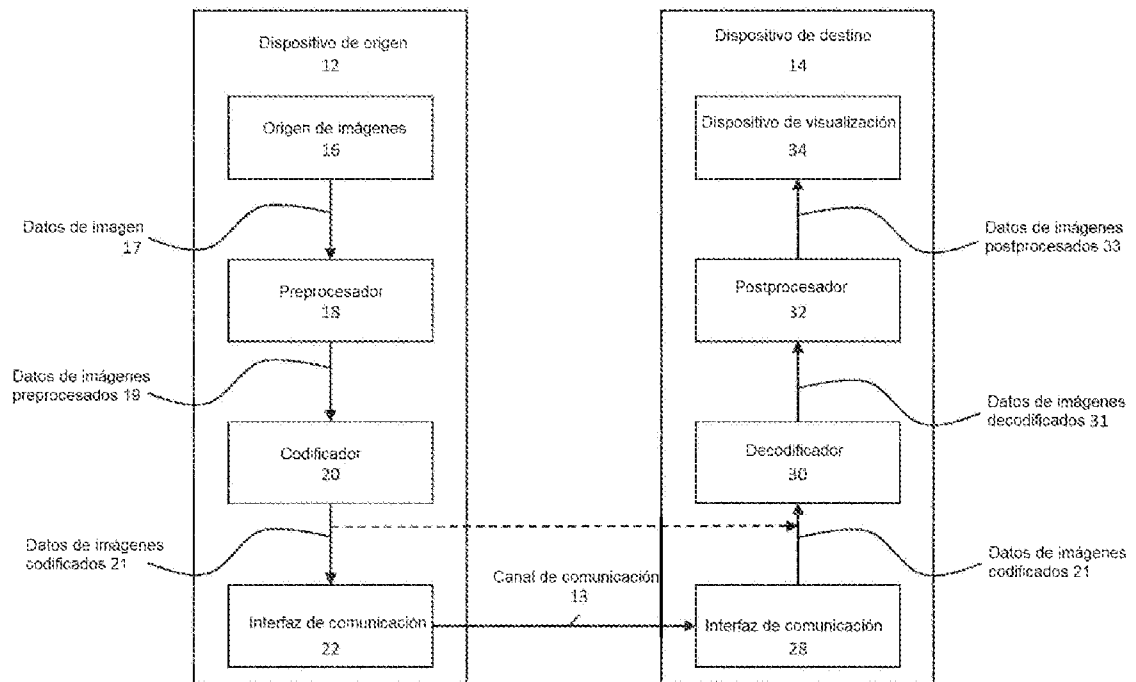
6. El método según la reivindicación 5, en donde el método comprende, además:

- 15 calcular una distancia de muestra para una muestra en el bloque actual;
calcular los factores de ponderación correspondientes a la muestra en función de la distancia de muestra calculada; y
- 20 calcular un valor de predicción combinado para la muestra según un primer valor de predicción, un segundo valor de predicción y los factores de ponderación.

7. Un codificador, que comprende:

- 25 uno o más procesadores; y
un medio de almacenamiento 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
- 30 decodificador para llevar a cabo el método de acuerdo con la reivindicación 5 o la reivindicación 6.

8. Un producto de programa informático que comprende un código de programa para realizar el método según una cualquiera de las reivindicaciones 1 a 3 y 5 a 6.



10

FIG. 1A

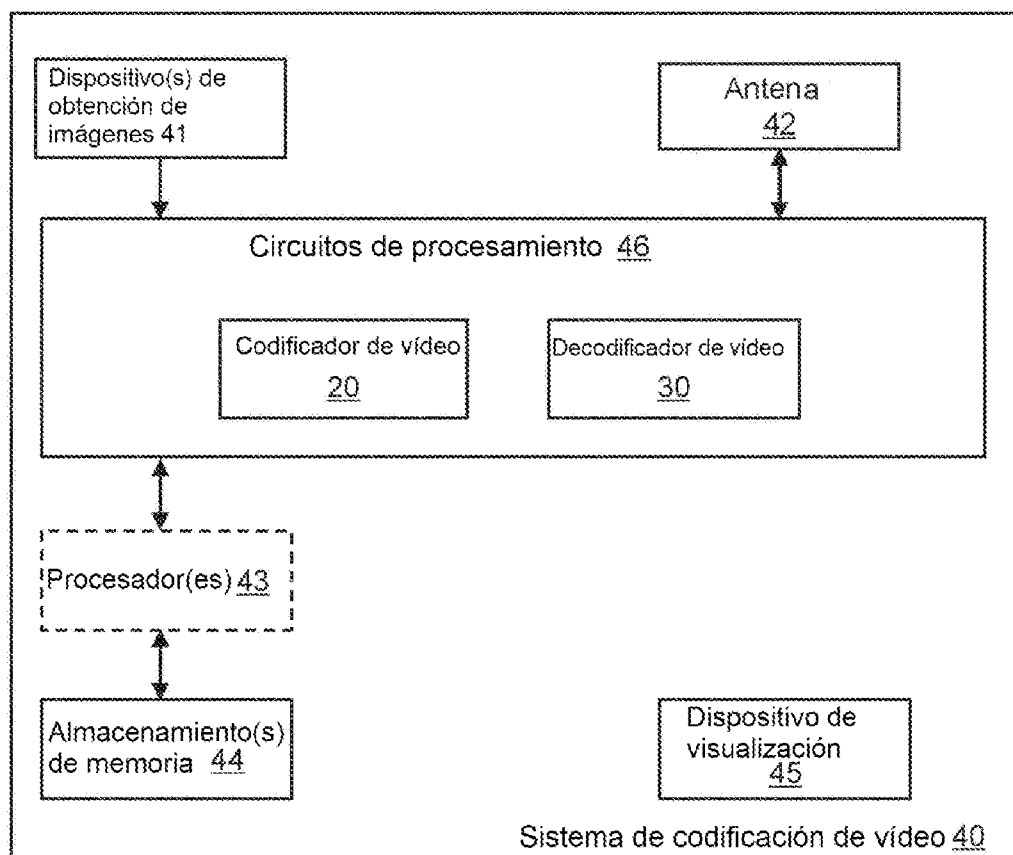


FIG. 1B

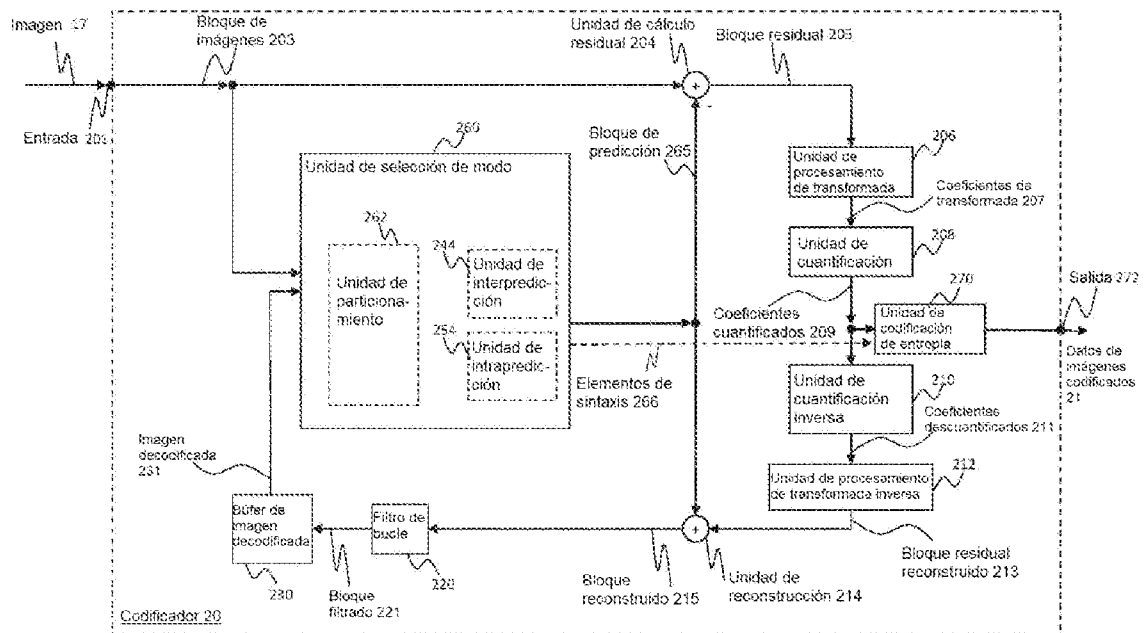


FIG. 2

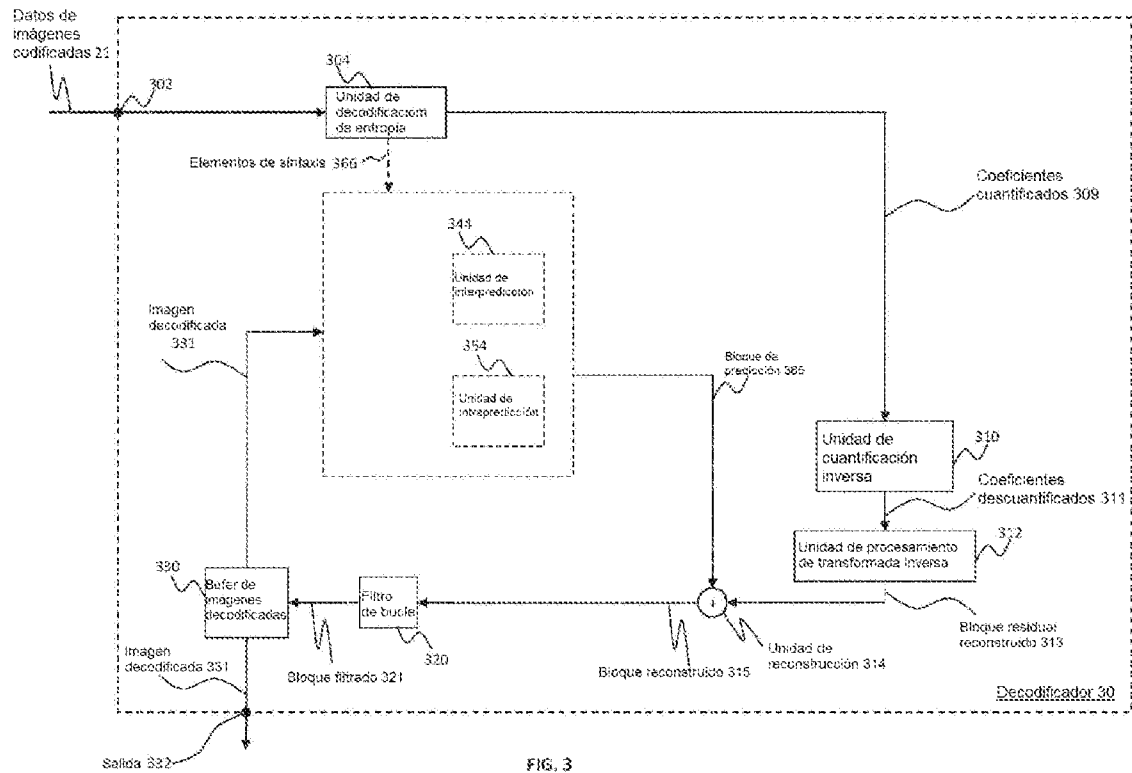


FIG. 3

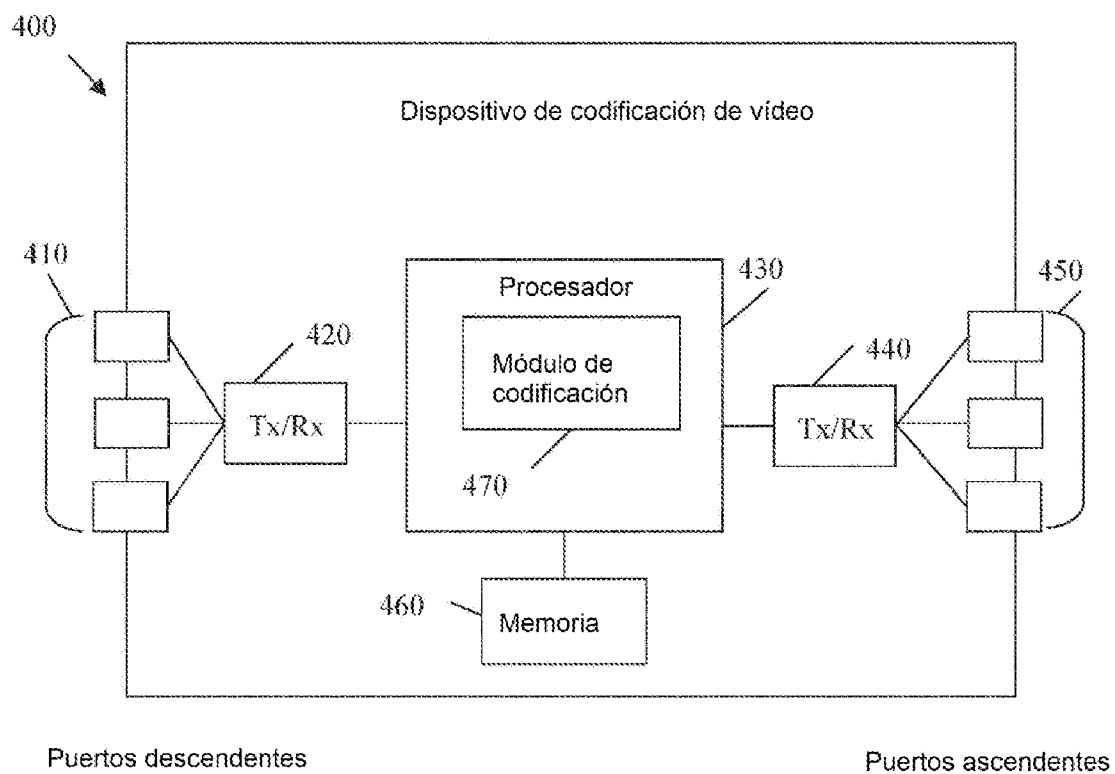


FIG. 4

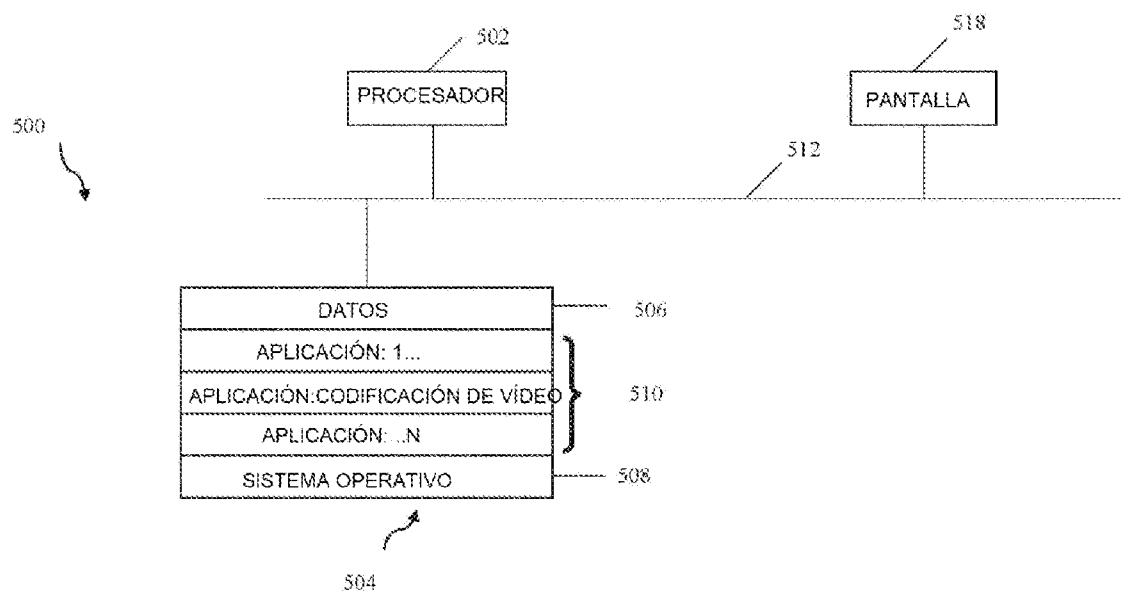


FIG. 5

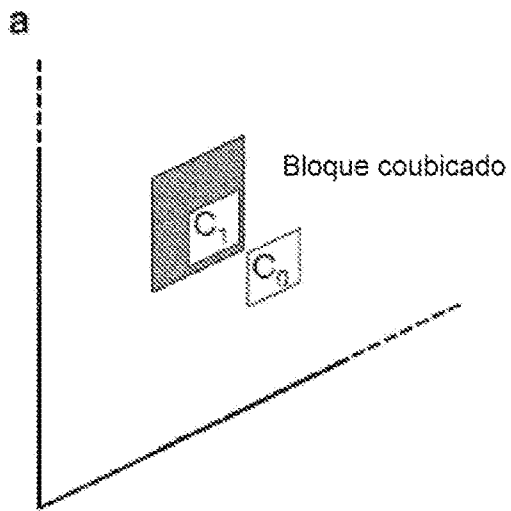


FIG. 6a

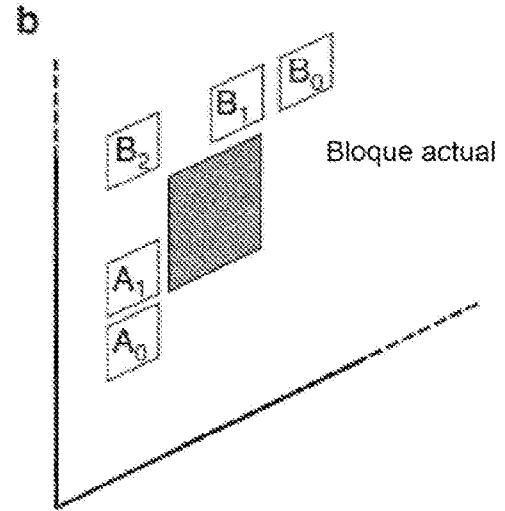
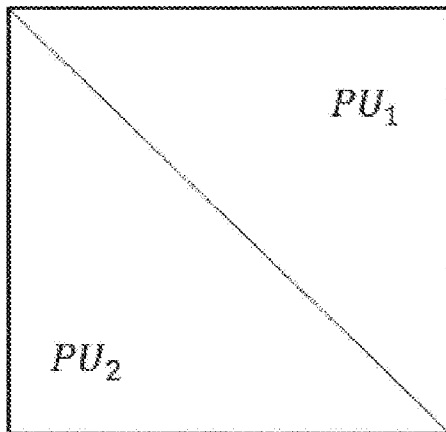
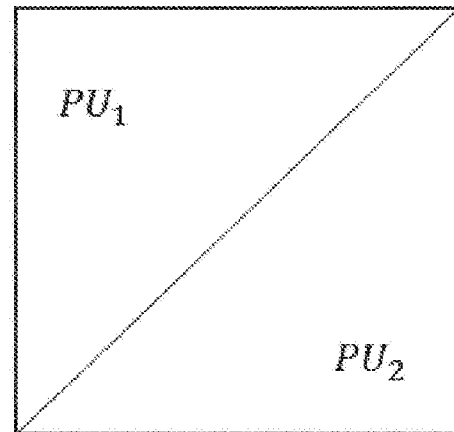


FIG. 6b



División desde la esquina superior izquierda a la esquina inferior derecha



División desde la esquina superior derecha a la esquina inferior izquierda

FIG. 7

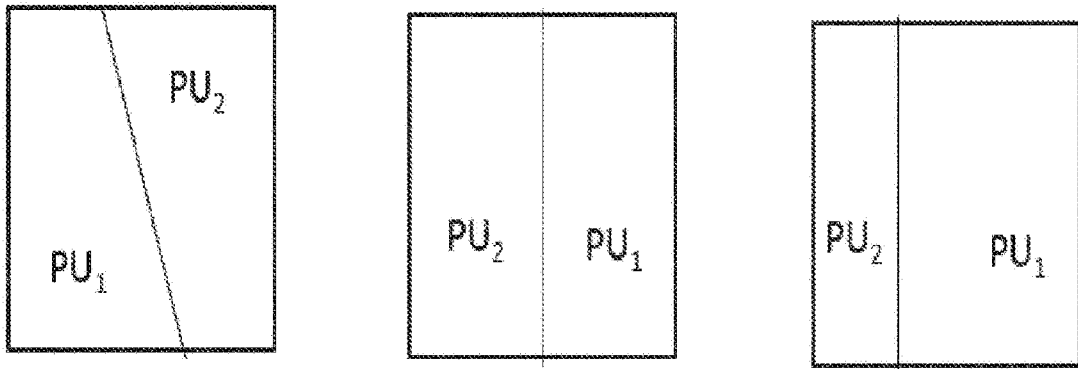


FIG. 8

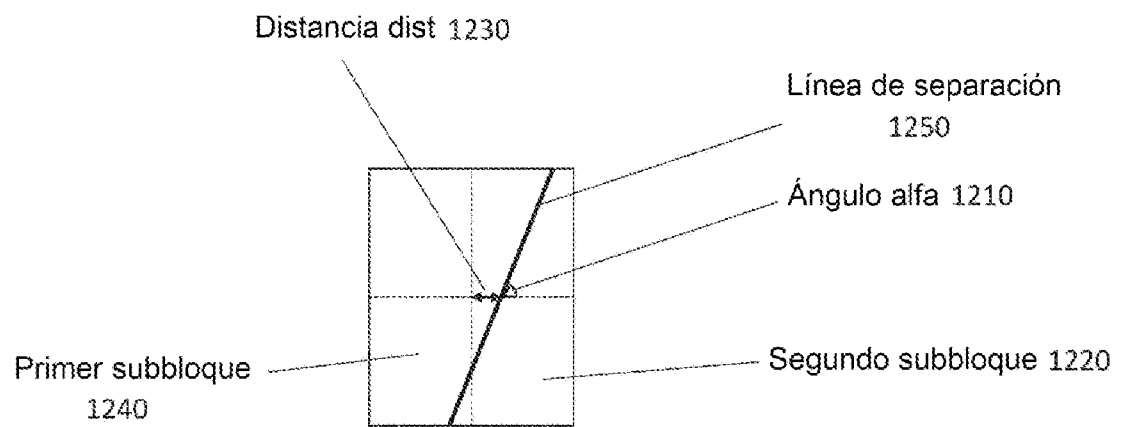


FIG. 9

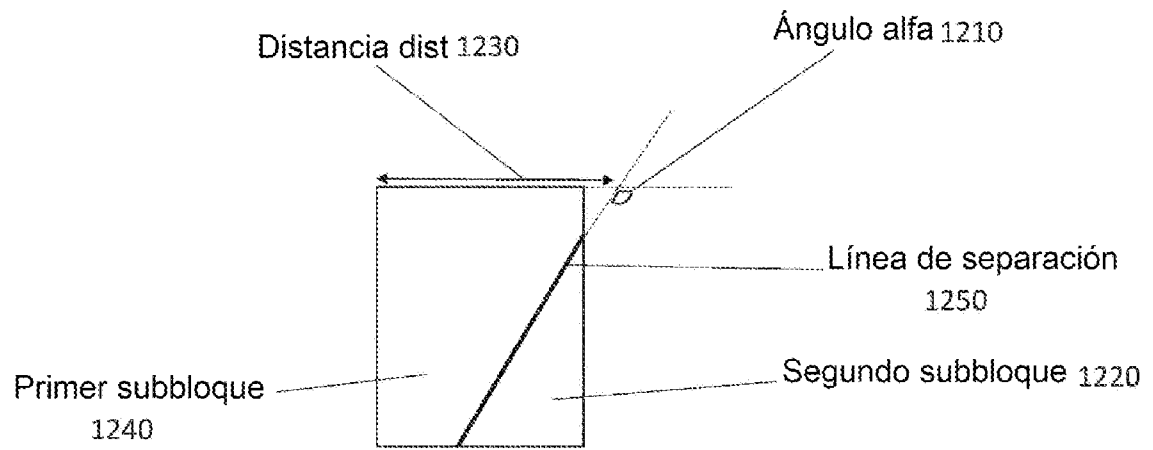


FIG. 10

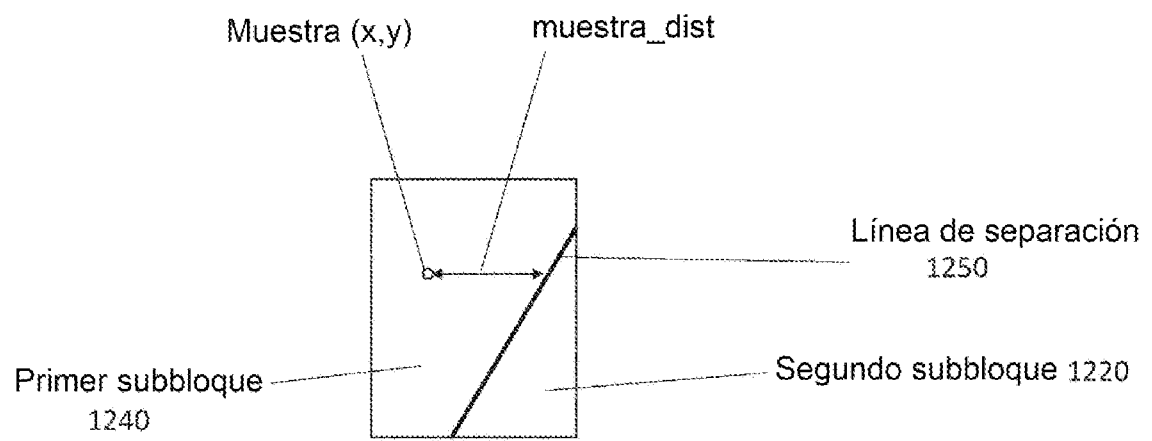


FIG. 11

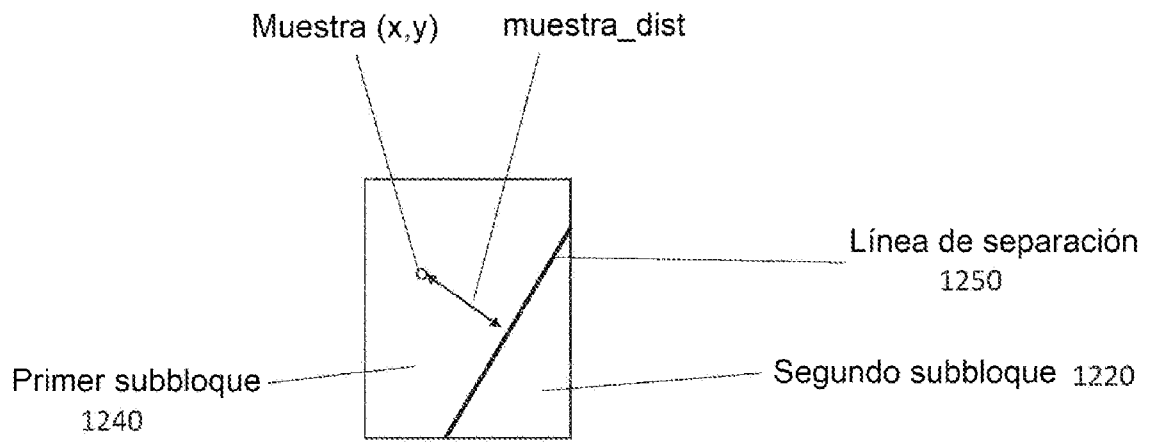


FIG. 12

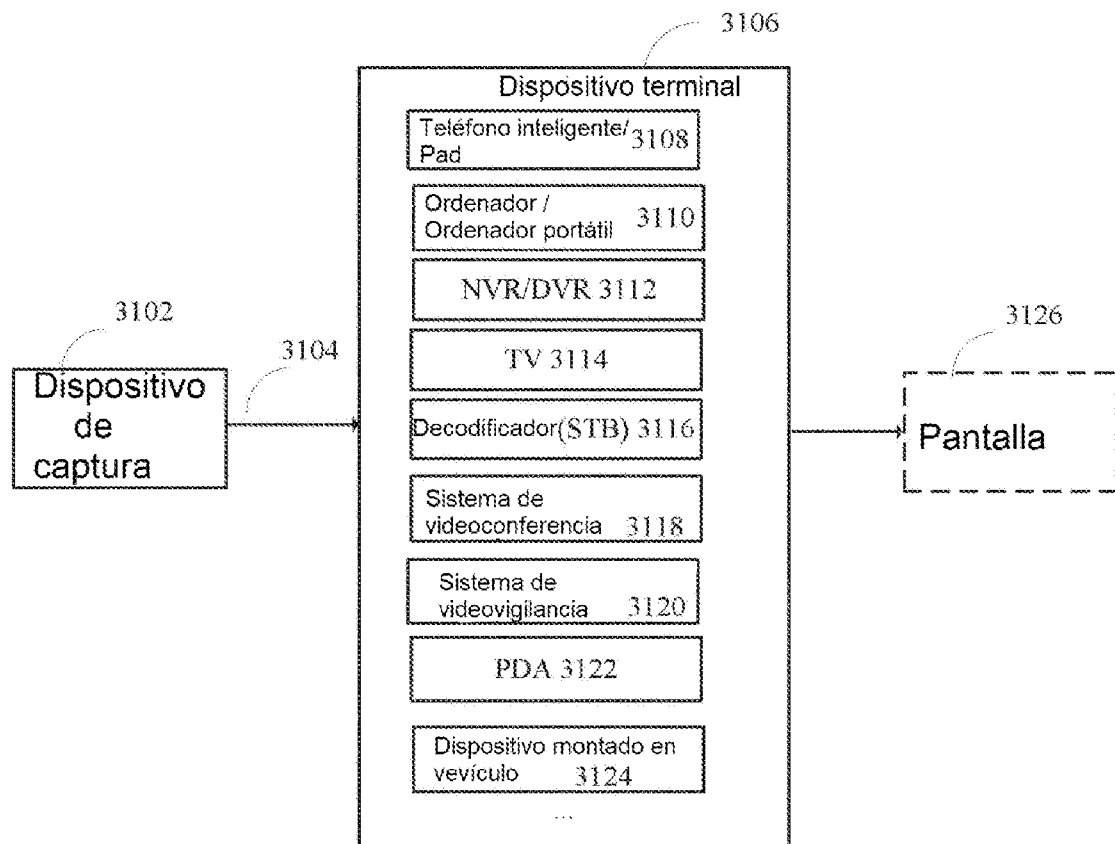


FIG. 13

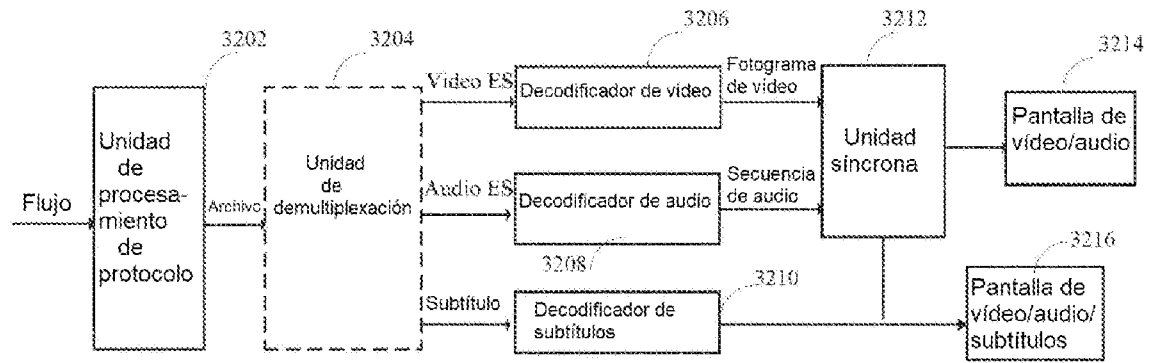


FIG. 14

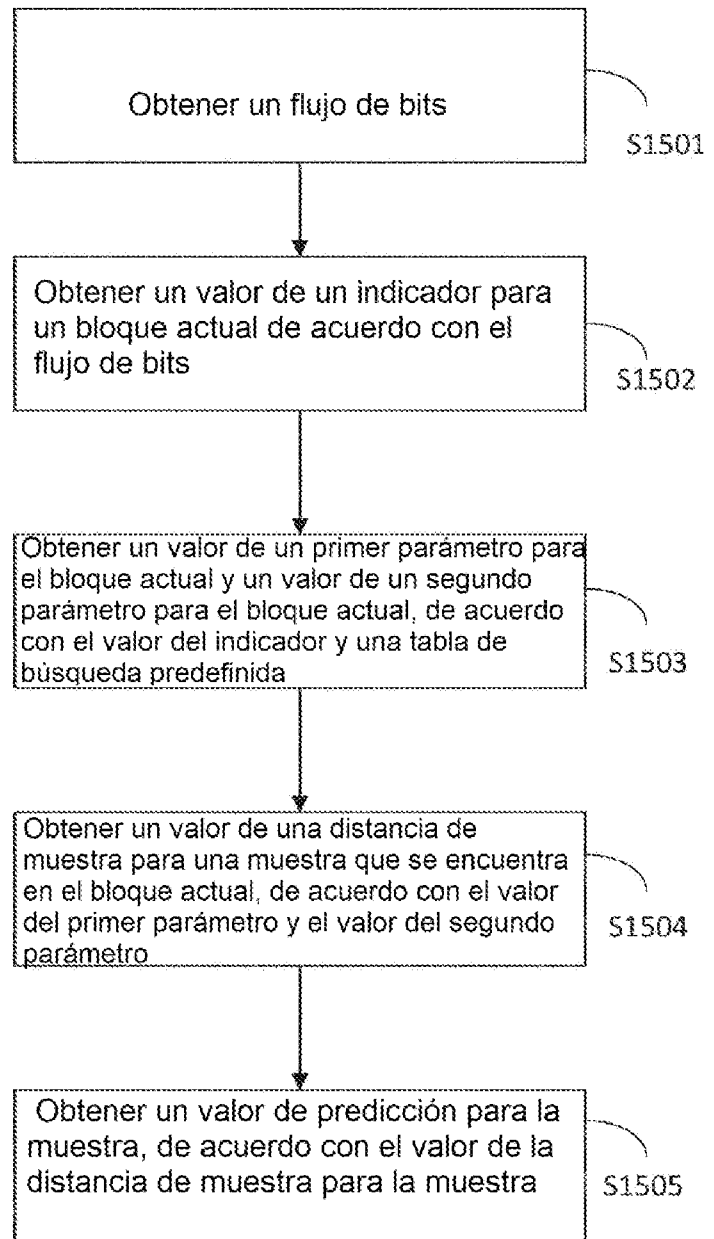


FIG.15

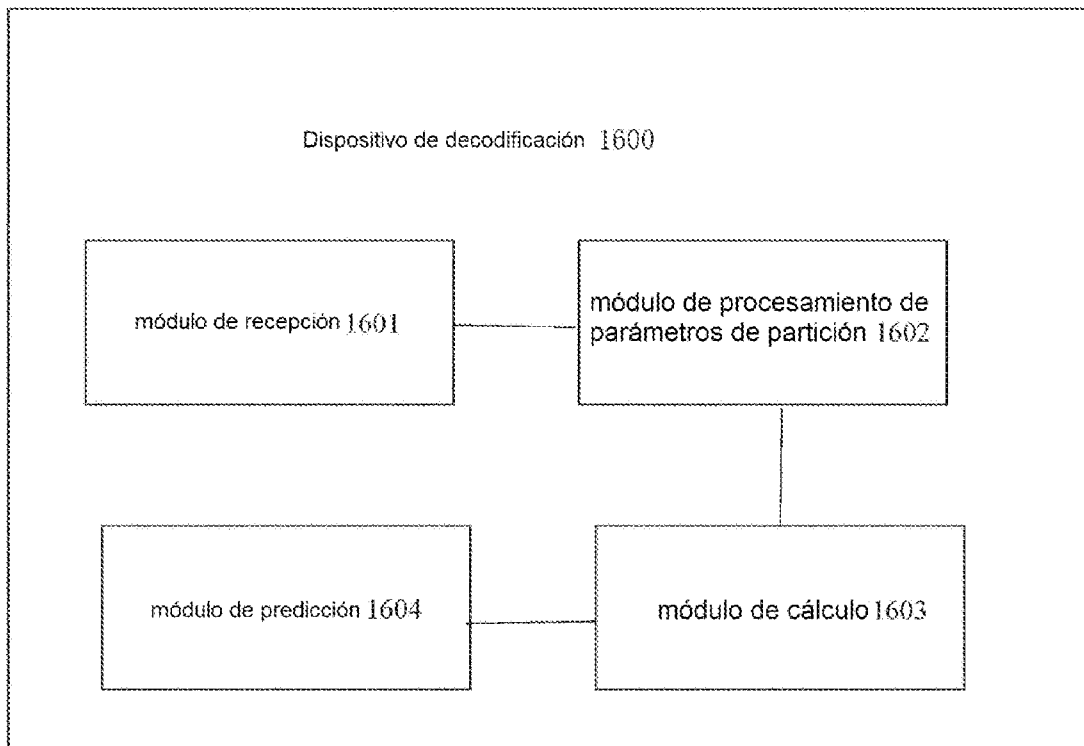


FIG.16