

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 994 578**

51 Int. Cl.:

H04N 19/124 (2014.01)
H04N 19/122 (2014.01)
H04N 19/186 (2014.01)
H04N 19/157 (2014.01)
H04N 19/70 (2014.01)
H04N 19/119 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **23.09.2020 PCT/US2020/052175**

87 Fecha y número de publicación internacional: **01.04.2021 WO21061744**

96 Fecha de presentación y número de la solicitud europea: **23.09.2020 E 20870350 (4)**

97 Fecha y número de publicación de la concesión europea: **04.09.2024 EP 4035368**

54 Título: **Procedimientos y dispositivos para diseño de cuantificación y descuantificación en codificación de vídeo**

30 Prioridad:
23.09.2019 US 201962904668 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
28.01.2025

73 Titular/es:
**BEIJING DAJIA INTERNET INFORMATION TECHNOLOGY CO., LTD. (100.00%)
Room 101D1-7, 1st Floor, Building 1 No.6,
Shangdi West Road Haidian District
Beijing 100085, CN**

72 Inventor/es:
**JHU, HONG-JHENG;
XIU, XIAOYU;
WANG, XIANGLIN;
CHEN, YI-WEN;
MA, TSUNG-CHUAN y
YU, BING**

74 Agente/Representante:
ELZABURU, S.L.P

ES 2 994 578 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

Procedimientos y dispositivos para diseño de cuantificación y descuantificación en codificación de vídeo

Referencia cruzada a la solicitud relacionada

5 Esta solicitud reivindica el beneficio de la solicitud provisional de EE. UU. Nº 62/904,668, presentada el 22 de septiembre de 2019.

Campo

La presente solicitud generalmente se relaciona con la codificación y compresión de vídeo, y en particular, pero no limitado a, procedimientos y dispositivos para diseño de cuantificación y descuantificación en codificación de vídeo.

Antecedentes

10 Se pueden utilizar diversas técnicas de codificación de vídeo para comprimir datos de vídeo. La codificación de vídeo se realiza de acuerdo con una o más normas de codificación de vídeo. Por ejemplo, las normas de codificación de vídeo incluyen codificación de vídeo versátil (VVC), modelo de prueba de exploración conjunta (JEM), codificación de vídeo de alta eficiencia (H.265/HEVC), codificación de vídeo avanzada (H.264/AVC), codificación del grupo de expertos en imágenes en movimiento (MPEG), o similares. La codificación de vídeo generalmente utiliza procedimientos de predicción (por ejemplo, interpredicción, intrapredicción o similares) que aprovechan la redundancia presente en las imágenes o secuencias de vídeo. Un objetivo importante de las técnicas de codificación de vídeo es comprimir datos de vídeo en una forma que utilice una tasa de bits más baja, mientras se evitan o minimizan las degradaciones en la calidad de vídeo.

20 El documento US10277910B2 se refiere a codificación de color de escape para modo de codificación de paleta. El documento SEREGIN (QUALCOMM) V ET AL: "Restriction on signalling for palette escape samples" 22. JCT-VC MEETING; GINEBRA; (EQUIPO COLABORATIVO CONJUNTO EN CODIFICACIÓN DE VÍDEO DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16), no. JCTVC-V0041 15 de octubre de 2015 se refiere al proceso de escalado para coeficientes de transformada.

Sumario

25 En general, esta divulgación describe ejemplos de técnicas relacionadas con el diseño de cuantificación y descuantificación en la codificación de vídeo. La invención se expone en el conjunto de las reivindicaciones adjuntas.

De acuerdo con un primer aspecto de la presente divulgación, se proporciona un procedimiento para descodificación de vídeo que incluye: derivar, a partir de un flujo de bits y para un primer bloque de codificación, un parámetro de cuantificación y un nivel cuantificado; derivar un valor de escala consultando una tabla de niveles de escala en base al parámetro de cuantificación y de acuerdo con una determinación de que el bloque de codificación está codificado en un modo de salto de transformada, obtener una muestra residual para una muestra en el primer bloque de codificación en base al nivel cuantificado, el valor de escala y una pluralidad de desplazamientos de bits mediante la aplicación de una primera operación de descuantificación, en donde la primera operación de descuantificación también se aplica para obtener una muestra reconstruida para una muestra de escape en un segundo bloque de codificación en un modo de paleta y en donde un tamaño de bloque del primer bloque de codificación o del segundo bloque de codificación no se utiliza para determinar parámetros de la primera operación de descuantificación, en donde la primera operación de descuantificación es la siguiente:

$$pSample = pLevel \times \frac{decScale[QP \% 6] \times 2^{(QP/6)}}{2^6}$$

40 en donde *pLevel* es un nivel cuantificado, *pSample* corresponde a un valor reconstruido de una muestra residual en el modo de salto de transformada o de una muestra de escape en el modo de paleta, *decScale*[*i*] es un valor de escala, *QP* es un parámetro de cuantificación, % designa una operación de módulo, *QP*%6 representa una operación del módulo *QP* 6.

45 De acuerdo con un segundo aspecto de la presente divulgación, se proporciona un dispositivo informático, que comprende: uno o más procesadores, un almacenamiento no transitorio acoplado al uno o más procesadores y una pluralidad de programas almacenados en el almacenamiento no transitorio que, cuando se ejecutan por el uno o más procesadores, hacen que el uno o más procesadores realicen el procedimiento mencionado anteriormente.

50 De acuerdo con un tercer aspecto de la presente divulgación, se proporciona un medio de almacenamiento legible por ordenador no transitorio que almacena un flujo de bits para su ejecución por un dispositivo informático que tiene uno o más procesadores, en donde el flujo de bits, cuando se ejecuta por uno o más procesadores, hace que el uno o más procesadores realicen el procedimiento mencionado anteriormente. De acuerdo con un cuarto aspecto de la presente divulgación, se proporciona un producto de programa informático, que comprende instrucciones almacenadas en el

mismo, en donde, cuando las instrucciones son ejecutadas por un procesador, las instrucciones hacen que el procesador realice el procedimiento mencionado anteriormente.

Breve descripción de las figuras

5 Se ofrecerá una descripción más particular de los ejemplos de la presente divulgación por referencia a ejemplos específicos ilustrados en los dibujos adjuntos. Dado que estos dibujos representan solo algunos ejemplos y, por lo tanto, no se consideran limitantes en su alcance, los ejemplos se describirán y explicarán con especificidad y detalles adicionales mediante el uso de los dibujos adjuntos.

La FIG. 1 es un diagrama en bloque que ilustra un codificador de video de ejemplo de acuerdo con algunas implementaciones de la presente divulgación.

10 La FIG. 2 es un diagrama en bloque que ilustra un decodificador de video de ejemplo de acuerdo con algunas implementaciones de la presente divulgación.

La FIG. 3 muestra una imagen dividida en unidades de árbol de codificación (CTU) de acuerdo con algunas implementaciones de la presente divulgación.

15 La FIG. 4 es un diagrama esquemático que ilustra modos de división de árbol de múltiples tipos de acuerdo con algunas implementaciones de la presente divulgación.

La FIG. 5 ilustra un mecanismo de señalización de información de división de partición en un árbol cuaternario con estructura de árbol de codificación de árbol de múltiples tipos anidados de acuerdo con algunas implementaciones de la presente divulgación.

20 La FIG. 6 muestra una CTU dividida en múltiples CU con un árbol cuaternario con estructura de bloque de codificación de árbol de múltiples tipos anidados de acuerdo con algunas implementaciones de la presente divulgación.

La FIG. 7 muestra un ejemplo de un bloque codificado en un modo de paleta de acuerdo con algunas implementaciones de la presente divulgación.

25 Las FIG. 8A y FIG. 8B muestran exploraciones transversales, horizontales y verticales de acuerdo con algunas implementaciones de la presente divulgación.

La FIG. 9 muestra la codificación de índices de paleta de acuerdo con algunas implementaciones de la presente divulgación.

La FIG. 10 es un diagrama en bloque que ilustra un aparato para codificación de video de acuerdo con algunas implementaciones de la presente divulgación.

30 La FIG. 11 es un diagrama en flujo que ilustra un proceso de ejemplo de diseño de cuantificación en codificación de video de acuerdo con algunas implementaciones de la presente divulgación.

La FIG. 12 es un diagrama de flujo que ilustra un proceso de ejemplo de diseño de descuantificación en codificación de video de acuerdo con algunas implementaciones de la presente divulgación.

Descripción detallada

35 Se hará ahora referencia en detalle a implementaciones específicas, ejemplos de las cuales se ilustran en los dibujos adjuntos. En la siguiente descripción detallada, se exponen numerosos detalles específicos no limitantes con el fin de ayudar a entender la materia objeto presentada en el presente documento. Pero será evidente para un experto en la técnica que pueden utilizarse diversas alternativas. Por ejemplo, será evidente para un experto en la técnica que la materia objeto presentada en el presente documento puede ser implementada en muchos tipos de dispositivos electrónicos con capacidades de video digital.

40 La referencia a lo largo de esta memoria descriptiva a "una realización", "una realización", "un ejemplo", "algunas realizaciones", "algunos ejemplos" o lenguaje similar significa que un rasgo, estructura o característica particular descrita se incluye en al menos una realización o ejemplo. Los rasgos, estructuras, elementos o características descritos en relación con una o algunas realizaciones también son aplicables a otras realizaciones, a menos que se especifique expresamente lo contrario.

45 A lo largo de la divulgación, los términos "primero", "segundo", "tercero", y etc. se utilizan todos como nomenclatura únicamente para referencias a elementos relevantes, por ejemplo, dispositivos, componentes, composiciones, pasos, y etc., sin implicar ningún orden espacial o cronológico, a menos que se especifique expresamente lo contrario. Por ejemplo, un "primer dispositivo" y un "segundo dispositivo" pueden hacer referencia a dos dispositivos formados de manera separada, o dos partes, componentes o estados operacionales de un mismo dispositivo, y pueden ser
50 nombrados arbitrariamente.

Los términos "módulo", "submódulo", "circuito", "subcircuito", "conjunto de circuitos", "subconjunto de circuitos", "unidad", o "subunidad" pueden incluir memoria (compartida, dedicada, o grupal) que almacena código o instrucciones que pueden ejecutarse por uno o más procesadores. Un módulo puede incluir uno o más circuitos con o sin código o instrucciones almacenados. El módulo o circuito puede incluir uno o más componentes que están directa o indirectamente conectados. Estos componentes pueden o no estar físicamente unidos, o ubicados adyacentes, entre sí. Como se utiliza en el presente documento, el término "si" o "cuando" pueden ser entendidos como "tras" o "en respuesta a" dependiendo del contexto. Estos términos, si aparecen en una reivindicación, puede no indicar que las limitaciones o características relevantes son condicionales u opcionales. Por ejemplo, un procedimiento puede comprender los pasos de: i) cuando o si la condición X está presente, se realiza la función o acción X', y ii) cuando o si la condición Y está presente, se realiza la función o acción Y'. El procedimiento puede ser implementado tanto con la capacidad de realizar la función o acción X', como con la capacidad de realizar la función o acción Y'. Por tanto, las funciones X' e Y' pueden ambas realizarse, en diferentes momentos, en múltiples ejecuciones del procedimiento.

Una unidad o módulo puede implementarse puramente mediante software, puramente mediante hardware, o mediante una combinación de hardware y software. En una implementación de software puro, por ejemplo, la unidad o módulo puede incluir bloques de código o componentes de software funcionalmente relacionados, que están directa o indirectamente enlazados entre sí, de manera que realice una función particular.

La FIG. 1 es un diagrama en bloque que ilustra un codificador de video de ejemplo de acuerdo con algunas implementaciones de la presente divulgación. En el codificador 100, un fotograma de video se particiona en una pluralidad de bloques de video para su procesamiento. Para cada bloque de video dado, se forma una predicción basada en un enfoque de interpredicción o un enfoque de intrapredicción. En la interpredicción, uno o más predictores se forman a través de estimación de movimiento y compensación de movimiento, en base a píxeles de los fotogramas reconstruidos previamente. En la intrapredicción, los predictores se forman en base a píxeles reconstruidos en una trama actual. A través de la decisión de modo, se puede elegir un mejor predictor para predecir un bloque actual.

Un residual de predicción, que representa la diferencia entre un bloque de video actual y su predictor, se envía a un circuito de transformada 102. Después, los coeficientes de transformada se envían desde el circuito de transformada 102 a un circuito de cuantificación 104 para la reducción de entropía. Los coeficientes cuantificados se alimentan entonces a un circuito de codificación de entropía 106 para generar un flujo de bits de video comprimidos. Como se muestra en la FIG. 1, la información relacionada con la predicción 110 de un circuito de interpredicción y/o de un circuito de intrapredicción 112, tal como información de la partición de bloques de video, vectores de movimiento, índice de imágenes de referencia y modo de intrapredicción, también se alimenta a través del circuito de codificación de entropía 106 y se guarda en un flujo de bits de video comprimidos 114.

En el codificador 100, también se necesitan conjuntos de circuitos relacionados con el decodificador para la reconstrucción de píxeles con fines de predicción. En primer lugar, se reconstruye un residual de predicción a través de una cuantificación inversa 116 y un circuito de transformada inversa 118. Este residual de predicción reconstruido se combina con un predictor de bloque 120 para generar píxeles reconstruidos sin filtrar para un bloque de video actual.

Para mejorar la eficiencia de la codificación y la calidad visual, se utiliza un filtro en bucle. Por ejemplo, un filtro de desbloqueo está disponible en AVC, HEVC, así como la versión actual de VVC. En HEVC, se define un filtro en bucle adicional denominado SAO (desplazamiento adaptivo de muestra) se define para mejorar además la eficiencia de codificación. En la versión actual de la norma VVC, otro filtro en bucle denominado ALF (filtro de bucle adaptivo) está siendo activamente investigado, y tiene buena oportunidad de ser incluido en la norma final.

La FIG. 2 es un diagrama en bloque que ilustra un decodificador 200 de video basado en bloque de ejemplo el cual puede utilizarse en conjunto con muchas normas de codificación de video. Este decodificador 200 es similar a la sección relacionada con la reconstrucción residente en el codificador 100 de la FIG. 1. En el decodificador 200, primero se decodifica un flujo de bits 201 de video entrante a través de una decodificación 202 por entropía para derivar niveles de coeficiente cuantificados e información relacionada con la predicción. Los niveles de coeficiente cuantificados se procesan después a través de una cuantificación 204 inversa y transformada 206 inversa para obtener un residual de predicción reconstruido. Un mecanismo predictor de bloques, implementado en un selector 212 de modo intra/inter, está configurado para realizar, bien una intrapredicción 208, o bien una compensación 210 de movimiento, en base a la información de predicción decodificada. Se obtiene un conjunto de píxeles reconstruidos sin filtrar sumando el residual de predicción reconstruido de la transformada 206 inversa y una salida predictiva generada por el mecanismo predictor de bloques, utilizando un sumador 214. En situaciones donde se activa el filtro en bucle 209, se realiza una operación de filtrado en estos píxeles reconstruidos para derivar una salida 222 de video reconstruida final.

Las normas de codificación/decodificación de video mencionadas anteriormente, tales como VVC, JEM, HEVC, MPEG-4, Parte 10, son conceptualmente similares. Por ejemplo, todas utilizan procesamiento basado en bloques. En una reunión del equipo conjunto de expertos en video (JVET), el JVET definió el primer borrador del método de codificación de Codificación de Vídeo Versátil (VVC) y el Modelo 1 de Prueba de VVC (VTM1). Se decidió incluir un árbol cuaternario con árbol de múltiples tipos anidados utilizando estructura de bloques de codificación de divisiones binarias y ternarias como la nueva característica de codificación inicial de VVC. Un árbol cuaternario es un árbol en el

que un nodo padre puede dividirse en cuatro nodos hijo, cada uno de los cuales puede convertirse en otro nodo padre para otra división en cuatro nuevos nodos hijo.

5 En VVC, la estructura de partición de imágenes divide el vídeo de entrada en bloques denominados unidades de árbol de codificación (CTU). Una CTU se divide utilizando un árbol cuaternario con estructura de árbol de múltiples tipos anidados en unidades de codificación (CU), con una CU hoja que define una región que comparte el mismo modo de predicción, por ejemplo, intra o inter. En la divulgación, el término "unidad" define una región de una imagen que cubre todos los componentes; el término "bloque" se usa para definir una región que cubre un componente particular, por ejemplo, luma, y puede diferir en la ubicación espacial cuando se considera el formato de muestreo de croma tal como 4:2:0.

10 Partición de una imagen en CTU

Una imagen puede dividirse en una secuencia de unidades de árbol de codificación (CTU). El concepto de CTU es el mismo que el de la HEVC. Para una imagen que tiene tres matrices de muestras, una CTU consiste en un bloque N×N de muestras de luma junto con dos bloques correspondientes de muestras de croma. La FIG. 3 muestra una imagen dividida en CTU de acuerdo con algunas implementaciones de la presente divulgación. El tamaño máximo permitido del bloque de luma en una CTU se especifica para que sea 128×128. Y el tamaño máximo de los bloques de transformada de luma es 64×64

Partición de CTU utilizando una estructura de árbol

20 En HEVC, una CTU se divide en CU utilizando una estructura de árbol cuaternario denominada árbol de codificación para adaptarse a diversas características locales. La decisión de si codificar un área de imagen utilizando predicción interimagen (temporal) o intraimagen (espacial) se toma a nivel de CU hoja. Cada CU de hoja puede dividirse adicionalmente en una, dos o cuatro unidades de predicción (PU) según un tipo de división de PU. Dentro de una PU, se aplica un mismo proceso de predicción y se transmite información relevante a un decodificador sobre una base de PU. Después de obtener un bloque residual aplicando el proceso de predicción basado en el tipo de división de PU, una CU hoja puede dividirse en unidades de transformada (TU) de acuerdo con otra estructura de árbol cuaternario similar al árbol de codificación para la CU. Una de las características clave de la estructura de HEVC es que tiene las múltiples concepciones de partición, incluidas CU, PU y TU.

30 En VVC, un árbol cuaternario con árbol de múltiples tipos anidados que utiliza una estructura de segmentación de divisiones binarias y ternarias reemplaza los conceptos de múltiples tipos de unidades de partición. Por ejemplo, elimina la separación de los conceptos de CU, PU y TU excepto cuando sea necesario para las CU que tengan un tamaño demasiado grande para la longitud de transformada máxima, y soporta más flexibilidad para las formas de partición de CU. En una estructura de árbol de codificación, una CU puede tener una forma cuadrada o rectangular. Una CTU se particiona primero mediante una estructura de árbol cuaternario, es decir, árbol cuaternario. A continuación, los nodos hoja del árbol cuaternario pueden particionarse además mediante una estructura de árbol de múltiples tipos.

35 La FIG. 4 es un diagrama esquemático que ilustra modos de división de árbol de múltiples tipos de acuerdo con algunas implementaciones de la presente divulgación. Como se muestra en la FIG. 4, hay cuatro tipos de división en la estructura de árbol de múltiples tipos, división binaria vertical (SPLIT_BT_VER), división binaria horizontal (SPLIT_BT_HOR), división ternaria vertical (SPLIT_TT_VER) y división ternaria horizontal (SPLIT_TT_HOR). Los nodos hoja del árbol de múltiples tipos se denominan CU, y a menos que la CU sea demasiado grande para la longitud de transformada máxima, esta segmentación se utiliza para la predicción y el procesamiento de transformada sin ninguna partición adicional. Por lo tanto, la CU, PU y TU pueden tener un mismo tamaño de bloque en el árbol cuaternario con estructura de bloque de codificación de árbol de múltiples tipos anidados. La excepción se produce cuando la longitud de transformada soportada máxima es menor que la anchura o altura del componente de color de la CU.

45 La FIG. 5 ilustra un mecanismo de señalización de información de división de partición en un árbol cuaternario con estructura de árbol de codificación de árbol de múltiples tipos anidados de acuerdo con algunas implementaciones de la presente divulgación. Una CTU se trata como una raíz de un árbol cuaternario y se particiona primero mediante una estructura de árbol cuaternario. Cada nodo hoja de árbol cuaternario, cuando es suficientemente grande para permitirlo, se particiona entonces además por una estructura de árbol de múltiples tipos. En la estructura de árbol de múltiples tipos, se señala un primer indicador (mtt_split_cu_flag) para indicar si el nodo está particionado adicionalmente. Cuando un nodo se particiona adicionalmente, se señala un segundo indicador (mtt_split_cu_vertical_flag) para indicar una dirección de división, y luego se señala un tercer indicador (mtt_split_cu_binary_flag) para indicar si la división es una división binaria o una división ternaria. En base a los valores de indicador vertical split_cu mtt y mtt_split_cu_binary_flag se deriva el modo de corte de árbol de múltiples tipos (MttSplitMode) de una CU tal como se muestra en la Tabla 1.

Tabla 1. Derivación MttSplitMode en base a elementos de sintaxis de árbol de múltiples tipos

MttSplitMode	mtt_split_cu_vertical_flag	mtt_split_cu-binary-flag
SPLIT_TT_HOR	0	0
SPLIT_BT_HOR	0	1
SPLIT_TT_VER	1	0
SPLIT_BT_VER	1	1

La FIG. 6 muestra una CTU dividida en múltiples CU con un árbol cuaternario con estructura de bloque de codificación de árbol de múltiples tipos anidados de acuerdo con algunas implementaciones de la presente divulgación. Como se muestra en la FIG. 6, los bordes de bloque en negrita representan la partición de árbol cuaternario y los bordes restantes representan la partición de árbol de múltiples tipos. El árbol cuaternario con partición de árbol de múltiples tipos anidada proporciona una estructura de árbol de codificación adaptativa al contenido compuesta por CU. El tamaño de una CU puede ser tan grande como la CTU o tan pequeño como 4x4 en unidades de muestras de luma. Para el caso del formato de croma 4:2:0, el tamaño máximo del bloque de codificación de croma (CB) es 64x64 y el tamaño mínimo del CB de croma es 2x2.

5 En VVC, el tamaño máximo de transformada de luma soportada es 64x64 y el tamaño máximo de transformada de croma soportada es 32x32. Cuando la anchura o altura de un CB es mayor que la anchura o altura de transformada máxima, el CB se divide automáticamente en la dirección horizontal y/o vertical para cumplir con la restricción de tamaño de transformada en esa dirección.

15 Los siguientes parámetros se definen y especifican por medio de los elementos de sintaxis del conjunto de parámetros de secuencia (SPS) para el árbol cuaternario con esquema de árbol de codificación de árbol de múltiples tipos anidados.

Tamaño de CTU: el tamaño de nodo raíz de un árbol cuaternario;

MinQTSIZE: el tamaño mínimo permitido de nodo hoja de árbol cuaternario;

MaxBtSize: el tamaño máximo permitido de nodo raíz de árbol binario;

20 MaxTtSize: el tamaño máximo permitido de nodo raíz de árbol ternario;

MaxMttDepth: la profundidad de jerarquía máxima permitida de la división de árbol de múltiples tipos desde un nodo hoja de árbol cuaternario;

MinBtSize: el tamaño mínimo permitido de nodo hoja de árbol binario;

MinTtSize: el tamaño mínimo permitido de nodo hoja de árbol ternario.

25 En un ejemplo del árbol cuaternario con estructura de árbol de codificación de árbol de múltiples tipos anidados, el tamaño de CTU se establece como 128x128 muestras de luma con dos bloques 64x64 correspondientes de muestras de croma 4:2:0. MinQTSIZE se establece como 16x16, MaxBtSize se establece como 128x128 y MaxTtSize se establece como 64x64. MinBtSize y MinTtSize, tanto para la anchura como para la altura, se establecen como 4x4, y MaxSieveSize se establece como 4. La partición de árbol cuaternario se aplica a la CTU primero para generar nodos hoja de árbol cuaternario. Los nodos hoja de árbol cuaternario pueden tener un tamaño de 16x16, es decir, MinQTSIZE, a 128x128, es decir, el tamaño de CTU. Si un nodo QT hoja es 128x128, no será dividido adicionalmente por el árbol binario puesto que el tamaño excede MaxBtSize y MaxTtSize, es decir, 64x64. De lo contrario, el nodo de árbol cuaternario hoja podría dividirse adicionalmente por el árbol de múltiples tipos. Por lo tanto, el nodo hoja de árbol cuaternario también es el nodo raíz para el árbol de múltiples tipos y tiene una profundidad de árbol de múltiples tipos, es decir mttDepth, como 0. Cuando la profundidad de árbol de múltiples tipos alcanza MaxMttDepth, es decir, 4, no se considera división adicional. Cuando el nodo de árbol de múltiples tipos tiene una anchura igual a MinBtSize y menor o igual a 2 * MinTtSize, no se considera división horizontal adicional. De manera similar, cuando el nodo de árbol de múltiples tipos tiene una altura igual a MinBtSize y menor o igual a 2 * MinTtSize, no se considera división vertical adicional.

40 La FIG. 7 muestra un ejemplo de un bloque codificado en un modo de paleta de acuerdo con algunas implementaciones de la presente divulgación. Para permitir el diseño de bloque de luma de 64x64 y de procesamiento en paralelo de croma de 32x32 en decodificadores de hardware VVC, la división TT está prohibida cuando o bien la anchura o la altura de un bloque de codificación luma es mayor que 64, como se muestra en la FIG. 7. La división TT también está prohibida cuando o bien la anchura o la altura de un bloque de codificación croma sea mayor que 32.

45 En VVC, el esquema de árbol de codificación soporta la capacidad de luma y croma para tener una estructura de árbol de bloques separada. Esta estructura de división de CU se denomina estructura de árbol dual o estructura de árbol de

codificación dual. La estructura de división de CU compartida tanto por luma como croma se denomina estructura de árbol único o estructura de árbol de codificación único. Para los cortes P y B, los bloques de árbol de codificación de luma y croma (CTB) en una CTU tienen que compartir la misma estructura de árbol de codificación. Sin embargo, para los cortes I, la luma y el croma pueden tener estructuras de árbol de bloques separadas. Cuando se aplica el modo de árbol de bloques separado, la CTB de luma se particiona en CU por una estructura de árbol de codificación, y las CTB de croma se particionan en CU de croma por otra estructura de árbol de codificación. Esto significa que una CU en un corte I puede consistir en un bloque de codificación del componente luma o bloques de codificación de dos componentes croma, y una CU en un corte P o B siempre consiste en bloques de codificación de los tres componentes de color a menos que el vídeo sea monocromático.

10 Selección de transformada múltiple (MTS) para transformada de núcleo

Además de la DCT-II que se ha empleado en la HEVC, se utiliza un esquema MTS para la codificación residual tanto de bloques intercodificados como intracodificados. Utiliza múltiples transformadas seleccionadas de DCT8/DST7. Las matrices de transformada recién introducidas son DST-VII y DCT-VIII. La Tabla 2 muestra funciones básicas de una DST/DCT seleccionada.

15 La Tabla 2 muestra funciones básicas de una DST/DCT seleccionada.

Tabla 2. Funciones de base de transformada de DCT-II/VIII y DST-VII para entrada de punto N

Tipo de transformada	Función base
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$ donde,
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \cos\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N + 2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (j + 1)}{2N + 1}\right)$

A fin de mantener la ortogonalidad de una matriz de transformada, las matrices de transformada se cuantifican con más precisión que las matrices de transformada en HEVC. Para mantener valores intermedios de coeficientes de transformada dentro del intervalo de 16 bits, después de la transformada horizontal y después de la vertical, todos los coeficientes deben tener 10 bits.

A fin de controlar el esquema MTS, se especifican indicadores de habilitación separados en el nivel de SPS a nivel SPS para intra e inter, respectivamente. Cuando el MTS está habilitado en el SPS, se señala un indicador de nivel de CU para indicar si el MTS está aplicado o no. Aquí, el MTS se aplica únicamente para luma. El indicador de nivel de CU MTS se señala cuando se satisfacen las siguientes condiciones: primero, tanto la anchura como la altura son menores o iguales a 32; segundo, el indicador CBF es igual a uno.

Si MTU_CU_flag es igual a cero, entonces se aplica DCT2 en ambas direcciones. Sin embargo, si MTS_CU_flag es igual a uno, entonces se señalan además otros dos indicadores para indicar el tipo de transformada para las direcciones horizontal y vertical, respectivamente. Tabla de transformada y mapeo de señalización como se muestra en la Tabla 3. Unificada la selección de transformada para ISP y MTS implícito se utiliza eliminando las dependencias intramodo y de forma de bloque. Si el bloque actual es el modo ISP o si el bloque actual es intrabloque y tanto el MTS explícito inter o intra están encendidos, entonces solo se utiliza DST7 para tanto núcleos de transformada horizontal como vertical. Cuando se trata de la precisión de la matriz de transformada, se utilizan núcleos de transformada primarios de 8 bits. Por lo tanto, todos los núcleos de transformada utilizados en HEVC se mantienen iguales, incluyendo DCT2 de 4 puntos y DST7, DCT2 de 8 puntos, 16 puntos y 32 puntos. Además, otros núcleos de transformada que incluyen DCT2 de 64 puntos, DCT8 de 4 puntos, DST7 y DCT8 de 8 puntos, 16 puntos, 32 puntos, utilizan núcleos de transformada primaria de 8 bits.

Tabla 3. Tabla de transformada y mapeo de señalización

MTS_CU_flag	MTS_Hor_Hag	MTS_Ver_flag	Intra/inter	
			Horizontal	Vertical
0			DCT2	
1	0	0	DST7	DST7
	0	1	DCT8	DST7
	1	0	DST7	DCT8
	1	1	DCT8	DCT8

Para reducir la complejidad de DST7 y DCT8 de gran tamaño, los coeficientes de transformada de alta frecuencia se ponen a cero para los bloques DST7 y DCT8 con un tamaño (anchura o altura, o tanto anchura como altura) igual a 32. Sólo se retienen los coeficientes dentro de la región de frecuencia más baja 16×16.

5 Como en HEVC, el residual de un bloque puede codificarse con el modo de salto de transformada. Para evitar la redundancia de la codificación de sintaxis, el indicador de salto de transformada no se señala cuando el nivel de CU CU_MTS_flag no es igual a cero. La limitación del tamaño de bloque para el salto de transformada es la misma que para MTS en JEM4, lo que indica que el salto de transformada es aplicable para una CU cuando tanto la anchura como la altura de bloque son iguales o menores que 32. Obsérvese que la transformada de MTS implícita se establece en DCT2 cuando se activa LFNST o MIP para la CU actual. Además, el MTS implícito puede todavía habilitarse cuando el MTS esté habilitado para bloques intercodificados.

Control de parámetros de cuantificación

15 En algunos ejemplos, el QP máximo puede extenderse de 51 a 63, y la señalización del parámetro de cuantificación (QP) inicial puede cambiarse en consecuencia. El valor inicial de SliceQpY se modifica en una capa de segmento de corte cuando se codifica un valor distinto de cero de slice_qp_delta. Específicamente, el valor de init_qp_minus26 puede modificarse para que esté en el intervalo de (-26 + QpBdOffsetY) a +37. Cuando el tamaño de un bloque de transformada no es una potencia de 4, los coeficientes de transformada se procesan junto con una modificación en la tabla de escala de nivel de QP o QP en lugar de mediante multiplicación por 181/256 (o 181/128), para compensar un escalado implícito por el proceso de transformada. Para el bloque de salto de transformada, el QP mínimo permitido se define como 4 porque el tamaño del paso de cuantificación se convierte en 1 cuando QP es igual a 4.

25 En algunos ejemplos, se utiliza una tabla de consulta fija para convertir un parámetro de cuantificación de luma QPY en un parámetro de cuantificación de croma QPC. En WC, se utiliza un mapeo QP luma a croma más flexible. En lugar de tener una tabla fija, la relación de mapeo QP luma a croma se señala en SPS utilizando un modelo lineal por tramos flexible, siendo la única restricción en el modelo lineal que la pendiente de cada tramo no puede ser negativa. Es decir, a medida que aumenta el QP de luma, el QP de croma debe permanecer plano o aumentar, pero no puede disminuir. El modelo lineal por tramos se define por: 1) número de tramos en el modelo; y 2) QP delta de entrada (luma) y salida (croma) para ese tramo. El intervalo de entrada del modelo lineal por tramos es [-QpBdOffsetY, 63] y el intervalo de salida del modelo lineal por tramos es [-QpBdOffsetC, 63]. La relación de mapeo QP puede señalizarse por separado para la codificación de Cb, Cr y Cb/Cr conjunta, o señalizarse conjuntamente para los tres tipos de codificación residual.

35 Al igual que en HEVC, se permite la adaptación de QP a nivel de CU en VVC. Los valores QP delta para componentes de luma y croma pueden señalizarse por separado. Para los componentes de croma, los valores de desplazamiento QP de croma permitidos se señalizan en forma de listas de desplazamiento en el conjunto de parámetros de imagen (PPS) de una manera similar a la de HEVC. Las listas se definen por separado para la codificación de Cb, Cr y Cb/Cr conjunta. Se permiten hasta 6 valores de desplazamiento para cada una de las listas de Cb, Cr y Cb/Cr juntas. A nivel de CU, se señala un índice para indicar cuál de los valores de desplazamiento en la lista de desplazamientos se utiliza para ajustar el QP de croma para esa CU.

Procesos de cuantificación y descuantificación para coeficientes de transformada

40 Los procesos de transformación requieren que los coeficientes resultantes se ajusten a escala en un cierto factor, seguido de una operación de desplazamiento durante los procesos de cuantificación y descuantificación. El factor de escala se define como sigue:

$$factor = \frac{1}{\sqrt{(M \cdot N)}} \tag{1}$$

donde M y N son la anchura y altura de un bloque de transformada.

En algunos ejemplos, las dimensiones de un bloque son potencias de 2, es decir, $M=2^m$ y $N=2^n$. Esto significa que cuando M es igual a N, o de hecho cuando $M \cdot N$ es potencia de 4, el factor puede aplicarse mediante un desplazamiento hacia la derecha. Los bloques que satisfacen tales condiciones se denominan "bloque normal". Cuando $M \cdot N$ no es una potencia de 4, se utilizan diferentes valores de escala y desplazamiento para compensar. Los valores de escala se definen como en la Tabla 4. Los bloques que satisfacen tales condiciones se denominan "bloque compensado".

5

Tabla 1. Tabla de escala

% QP 6		0	1	2	3	4	5
encScale[QP%6]	Bloques normales	26214	23302	20560	18396	16384	14564
decScale[QP%6]		40	45	51	57	64	72
encScale[QP%6]	Bloques compensados	18396	16384	14564	13107	11651	10280
decScale[QP%6]		57	64	72	80	90	102

Para bloques codificados en modo de salto de transformada, se realizan operaciones de escala y desplazamiento definidas para bloques normales.

La derivación de un coeficiente de transformada escalado se ilustra en la Tabla 5. Las definiciones de todas las variables en la Tabla 5 pueden encontrarse en la versión 6 de la especificación de borrador de VVC.

10

Tabla 5

<p>La variable <code>rectNonTsFlag</code> se deriva como sigue: <code>rectNonTsFlag = (((Log2(nTbW) + Log2(nTbH)) & 1) == 1 && transform_skip_flag[xTbY][yTbY] == 0)</code> Las variables <code>bdShift</code>, <code>rectNorm</code> y <code>bdOffset</code> se derivan como sigue: <code>bdShift = bitDepth + ((rectNonTsFlag ? 1 : 0) + (Log2(nTbW) + Log2(nTbH)) / 2) - 5 + dep_quant_enabled_flag</code> <code>bdOffset = (1 << bdShift) >> 1</code></p> <p>La lista <code>levelScale[][]</code> se especifica como <code>levelScale[j][k] = { { 40, 45, 51, 57, 64, 72 }, { 57, 64, 72, 80, 90, 102 } }</code> con $j = 0, 1$, $k = 0, 5$.</p> <p>La matriz $(nTbW) \times (nTbH)$ <code>dz</code> se establece igual a la matriz $(nTbW) \times (nTbH)$ <code>TransCoeffLevel[xTbY][yTbY][cIdx]</code>. Para la derivación de los coeficientes de transformada escalados <code>d[x][y]</code> con $x = 0..nTbW - 1$, $y = 0..nTbH - 1$, se aplica lo siguiente:</p> <ul style="list-style-type: none"> - El factor de escalado intermedio <code>m[x][y]</code> se deriva como sigue: <ul style="list-style-type: none"> - Si una o más de las siguientes condiciones son verdaderas, <code>m[x][y]</code> se establece igual a 16: <ul style="list-style-type: none"> - <code>sps_scaling_list_enabled_flag</code> es igual a 0. - <code>transform_skip_flag[xTbY][yTbY]</code> es igual a 1. - De lo contrario, se aplica lo siguiente: <code>m[x][y] = ScalingFactor[Log2(nTbW)][Log2(nTbH)][matrixId][x][y]</code> - El factor de escalado <code>1s[x][y]</code> se deriva como sigue: <ul style="list-style-type: none"> - Si <code>ep_quant_enabled_flag</code> es igual a 1, se aplica lo siguiente: <code>1s[x][y] = (m[x][y] * levelScale[rectNonTsFlag][(qP + 1) % 6]) << ((qP + 1) / 6)</code> - De lo contrario (<code>dep_quant_enabled_flag</code> es igual a 0), se aplica lo siguiente: <code>1s[x][y] = (m[x][y] * levelScale[rectNonTsFlag][qP % 6]) << (qP / 6)</code> - Cuando <code>BdpcmFlag[xTbY][yYbY]</code> es igual a 1, <code>dz[x][y]</code> se modifica como sigue: <ul style="list-style-type: none"> - Si <code>BdpcmDir[xTbY][yYbY]</code> es igual a 0 y x es mayor que 0, se aplica lo siguiente: <code>dz[x][y] = Clip3(CoeffMin, CoeffMax, dz[x-1][y] + dz[x][y])</code> - De lo contrario, si <code>BdpcmDir[xTbY][yYbY]</code> es igual a 1 e y es mayor que 0, se aplica lo siguiente: <code>dz[x][y] = Clip3(CoeffMin, CoeffMax, dz[x][y-1] + dz[x][y])</code> - El valor <code>dnc[x][y]</code> se deriva como sigue: <code>dnc[x][y] = (dz[x][y] * 1s[x][y] + bdOffset) >> bdShift</code>
--

- El coeficiente de transformada escalado $d[x][y]$ se deriva como sigue:
 $d[x][y] = \text{Clip3}(\text{CoeffMin}, \text{CoeffMax}, \text{dnc}[x][y])$

Dado un valor QP utilizado, los procesos de cuantificación y descuantificación para coeficientes de transformada pueden describirse como sigue. La cuantificación puede describirse mediante la ecuación (2):

$$pLevel = \text{Quant}(pCoeff) = pCoeff \times \frac{encScale[rectNonTsFlag][QP\%6]}{2^{14} \times 2^{(QP/6)} \times 2^{transformShift'}} \quad (2)$$

La de cuantificación puede describirse mediante la ecuación (3):

$$pCoeff' = \text{Quant}^{-1}(pLevel) = pLevel \times \frac{decScale[rectNonTsFlag][QP\%6] \times 2^{(QP/6)} \times 2^{transformShift'}}{2^6} \quad (3)$$

donde la variable *rectNonTsFlag* puede obtenerse mediante la ecuación (4):

$$rectNonTsFlag = (((\text{Log2}(W) + \text{Log2}(H)) \& 1) == 1 \&\& transform_skip_flag == 0) \quad (4)$$

y la variable *transformShift'* puede obtenerse por la ecuación (5):

$$transformShift' = rectNonTsFlag ? (transformShift - 1) : transformShift \quad (5)$$

10 Aquí, *pCoeff* es un valor de coeficiente de transformada; *pLevel* es un valor cuantificado o nivel de cuantificación; *pcoeff'* son valores de coeficiente de transformada reconstruidos a partir del proceso de descuantificación.

La variable *rectNonTsFlag* representa si un bloque actual se clasifica como un "bloque normal" o un "bloque compensado". Cuando tiene un valor de falso o 0, el bloque actual se clasifica como un bloque normal. Cuando tiene un valor de verdadero o 1, el bloque actual se clasifica como un bloque compensado.

15 La variable *transformShift* representa un desplazamiento de bits que se utiliza para compensar el incremento de rango dinámico debido a transformadas bidimensionales (2D), el cual es igual a $15 - bitDepth - (\log_2(W) + \log_2(H))/2$, donde *W* y *H* son la anchura y altura de la unidad de transformada actual, *bitDepth* es una profundidad de bit de codificación. Dependiendo del valor de *rectNonTsFlag*, el valor actual utilizado en la operación de desplazamiento, *transformShift'*, puede tomar un mismo valor que *transformShift*, o un valor igual a (*transformShift*-1).

20 *encScale*[[]] y *decScale*[[]] son valores de escalado de cuantificación y descuantificación que están en precisión de 14 bits y 6 bits respectivamente, y se definen como se muestra en la Tabla 1. Dependiendo del valor de *rectNonTsFlag*, una serie de factores de escalado utilizados para los "bloques normales" o "bloques compensados" se seleccionan y utilizan para el bloque actual.

Procesos de cuantificación y descuantificación para el modo de salto de transformada

25 En un modo de salto de transformada, los residuales de predicción se cuantifican y codifican directamente sin operaciones de transformada realizadas. Más específicamente, sus procesos de cuantificación y descuantificación se pueden describir como sigue.

La cuantificación puede describirse por la ecuación (6):

$$pLevel = \text{Quant}_{TS}(pResi) = pResi \cdot 2^{transformShift} \frac{encScale[QP\%6]}{2^{14} \times 2^{(QP/6)} \times 2^{transformShift'}} \quad (6)$$

30 La descuantificación puede describirse por la ecuación (7):

$$pResi' = \text{Quant}_{TS}^{-1}(pLevel) = \frac{pLevel}{2^{transformShift}} \times \frac{decScale[QP\%6] \times 2^{(QP/6)} \times 2^{transformShift'}}{2^6} \quad (7)$$

donde *pResi* y *pResi'* son valores originales y reconstruidos de muestra residual de predicción; *pLevel* es un valor cuantificado, o nivel de cuantificación; *encScale*[[]] y *decScale*[[]] son los valores de escalado de cuantificación y

descuantificación que están en precisión de 14 bits y 6 bits respectivamente, y definidos los mismos que los utilizados para "bloques normales" mostrados en la Tabla 1.

Procesos de cuantificación y descuantificación para modo de paleta

5 VTM6 soporta el modo de paleta para codificación de contenido de pantalla en formato de color 4:4:4. Cuando el modo de paleta está habilitado y si un tamaño de CU es menor o igual a 64x64, se transmite un indicador al nivel de CU para indicar si el modo de paleta se utiliza para la CU. El modo de paleta se señala como un modo de predicción distinto de un modo de intrapredicción, interpredicción e intracopia de bloques (IBC).

10 Si el modo de paleta se utiliza para una CU, los valores de muestra en la CU pueden representarse por un pequeño conjunto de valores de color representativos. El conjunto se denomina "paleta". Para píxeles con valores cercanos a los colores de paleta, los índices de paleta pueden señalizarse para transmitir sus valores al decodificador. También es posible especificar una muestra cuyos valores no están cerca de ningún color de paleta señalizando un índice de símbolo de escape, seguido de valores de escape. Los valores de escape son los valores de componentes cuantificados de la muestra. Esto se ilustra en la FIG. 7.

15 Para la cuantificación y descuantificación del valor de escape, las siguientes ecuaciones describen los procesos correspondientes que se aplican en el codificador y el decodificador, respectivamente.

La cuantificación puede describirse mediante la ecuación (8):

$$pLevel = Quant_E(pEsca) = pEsca \times \frac{encScale[QP\%6]}{2^{14} \times 2^{(QP/6)}} \quad (8)$$

La descuantificación puede describirse mediante la ecuación (9):

$$pEsca' = Quant_E^{-1}(pLevel) = pLevel \times \frac{decScale[QP\%6] \times 2^{(QP/6)}}{2^6} \quad (9)$$

20 donde $pEsca$ y $pEsca'$ son los valores de escape reconstruidos; $pLevel$ es un valor cuantificado, y nivel de cuantificación; $encScale[]$ y $decScale[]$ son valores de escalado de cuantificación y descuantificación que están respectivamente en precisión de 14 bits y 6 bits, y se definen igual que los utilizados para "bloques normales" mostrados en la Tabla 4.

25 Para la codificación de la paleta, se mantiene un predictor de paleta que incluye una lista de colores. El predictor se inicializa a 0 (es decir, lista vacía) al comienzo de cada corte para el caso sin frente de onda y al comienzo de cada fila de CTU para el caso con frente de onda. Para cada entrada en el predictor de paleta, se señala un indicador de reutilización para indicar si es parte de la paleta actual en la CU. Los señalizadores de reutilización se envían utilizando codificación de longitud de ejecución de ceros. Después de esto, se señala el número de nuevas entradas de paleta y los valores de componente para las nuevas entradas de paleta. Después de codificar una CU en modo de paleta, el predictor de paleta se actualiza utilizando la paleta actual, y las entradas del predictor de paleta que no se reutilizan en la paleta actual se añaden al final hasta que se alcanza el tamaño máximo de paleta permitido para formar el nuevo predictor de paleta. Se señala un indicador de escape para cada CU para indicar si están presentes símbolos de escape en la CU actual. Si hay símbolos de escape presentes, la tabla de paleta se aumenta en uno y el último índice se asigna para representar el símbolo de escape.

35 La FIG. 8A y la FIG. 8B muestran exploraciones transversales horizontales y verticales de acuerdo con algunas implementaciones de la presente divulgación. Los índices de paleta de muestras en una CU forman un mapa de índices de paleta. El mapa de índices se codifica utilizando exploraciones transversales horizontales y/o verticales como se muestra en las FIG. 8A y 8B. El orden de exploración se señala explícitamente en el flujo de bits utilizando el `palette_transpose_flag`.

40 La FIG. 9 muestra codificación de índices de paleta de acuerdo con algunas implementaciones de la presente divulgación. Los índices de paleta son codificados utilizando dos modos de muestra de paleta principales: INDEX y COPY_ABOVE. El modo es señalado utilizando un indicador excepto para la fila superior donde únicamente la exploración horizontal se utiliza, la primera columna donde únicamente la exploración vertical se utiliza, o para ubicaciones de muestra de paleta donde el modo anterior fue COPY_ABOVE. En el modo COPY_ABOVE, el índice de paleta de la muestra en la fila superior es copiado. En el modo de INDEX, el índice de paleta es explícitamente señalado. Tanto para los modos INDEX como COPY_ABOVE, se señala un valor de serie que especifica el número de píxeles que se codifican utilizando el mismo modo asociado.

50 El orden de codificación para el mapa de índices es el siguiente. En primer lugar, se señala el número de valores de índice asociados a ejecuciones de INDEX. Esto va seguido de la señalización de los valores de índice reales para toda la CU utilizando codificación binaria truncada. A continuación, el modo de paleta (INDEX o COPY_ABOVE) y la longitud de ejecución para cada ejecución se señalizan de manera intercalada. Finalmente, los colores cuantificados en modo escape para toda la CU se agrupan y codifican con codificación exponencial de Golomb.

Para los cortes con árbol de luma/croma dual, la paleta se aplica sobre luma (componente Y) y croma (componentes Cb y Cr) por separado. Para los cortes de un árbol individual, la paleta se aplicará en los componentes Y, Cb, Cr conjuntamente, es decir, cada entrada en la paleta contiene valores Y, Cb, Cr. Para el desbloqueo, el límite de bloque de un bloque codificado por paleta no se desbloquea.

5 Tres diseños diferentes de esquema de cuantificación están disponibles y se aplican a la transformada regular, salto de transformada y modo de paleta, respectivamente. Cada diseño de cuantificación diferente está asociado con operaciones de desplazamiento y escala diferentes. Para bloques en los que se aplica la transformada regular, las operaciones de desplazamiento y escala dependen de la forma del bloque. Para bloques en los que se aplica salto de transformada, las operaciones de desplazamiento y escala no dependen de la forma del bloque. Para bloques codificados en modo paleta, solo se realiza la operación de escala, y la operación no depende de la forma del bloque. Tal diseño no unificado puede no ser óptimo desde el punto de vista de la normalización.

En algunos ejemplos, los procedimientos se proporcionan para simplificar y mejorar además la transformada y la cuantificación. Las operaciones de cuantificación y descuantificación utilizadas en el modo de salto de transformada y el modo de paleta pueden unificarse aplicando las operaciones de cuantificación y descuantificación utilizadas en el modo de salto de transformada al modo de paleta.

En algunos ejemplos, los métodos se proporcionan para simplificar y mejorar además la transformada y cuantificación. Las operaciones de cuantificación y descuantificación utilizadas bajo modo de salto de transformada y modo de paleta pueden ser unificados aplicando operaciones de cuantificación y descuantificación utilizados bajo el modo de salto de transformada al modo de paleta.

20 En algunos ejemplos, las operaciones de cuantificación y descuantificación del modo de transformada regular, el modo de salto de transformada y el modo de paleta se unifican todas aplicando las operaciones de cuantificación y descuantificación para el modo de transformada regular a todos los modos, incluyendo también el modo de salto de transformada y el modo de paleta.

Armonización de cuantificación y descuantificación utilizada para modo de salto de transformada y modo de paleta

25 En algunos ejemplos, un proceso de cuantificación y descuantificación se aplica tanto a residuales en el modo de salto de transformada y valores de escape en el modo de paleta. Las operaciones de cuantificación y descuantificación que se utilizan en el modo de salto de transformada se aplican al modo de paleta también. Por ejemplo, las ecuaciones (6) y (7) correspondientes a $Quant_{TS}(pResi)$ y $Quant_{TS}^{-1}(pLevel)$ se utilizan para valores de escape en modo paleta, en

donde $pResi$ se sustituiría por los valores de escape originales $pEsca$, y la salida de $Quant_{TS}^{-1}(pLevel)$ en este caso serían los valores de escape reconstruidos $pEsca'$. Específicamente, la operación de cuantificación se describe como:

$$pLevel = pEsca \times 2^{transformShift} \frac{encScale[QP\%6]}{2^{14} \times 2^{(QP/6)} \times 2^{transformShift}}$$

donde $transformShift = 15 - bitDepth - (\log_2(W) + \log_2(H))/2$, $pLevel$ es el nivel cuantificado, $pEsca$ es el valor del color de escape, QP es el parámetro de cuantificación, $encScale[]$ es el valor de escala, % denota una operación de módulo, $QP\%6$ representa una operación de módulo $QP/6$, W es una anchura de la CU, H es una altura de la CU, y $bitDepth$ es una profundidad de bits de codificación.

Además, la operación de descuantificación se describe, por consiguiente, como:

$$pEsca' = \frac{pLevel}{2^{transformShift}} \times \frac{decScale[QP\%6] \times 2^{(QP/6)} \times 2^{transformShift}}{2^6}$$

donde $transformShift = 15 - bitDepth - (\log_2(W) + \log_2(H))/2$, $pLevel$ es el nivel cuantificado, $pEsca'$ es el valor de color de escape reconstruido para la CU, el QP es el parámetro de cuantificación, $decScale[]$ es el valor de escala, % designa la operación del módulo, $QP\%6$ representa la operación del módulo $QP/6$, W es la anchura de la CU, H es la altura de la CU, y $bitDepth$ es la profundidad de bits de codificación.

En algunos ejemplos, las operaciones de cuantificación y descuantificación que se utilizan para valores de escape bajo el modo de paleta se aplican a residuales de predicción en el modo de salto de transformada también. Por

ejemplo, las ecuaciones (8) y (9) correspondientes a $Quant_E(pEsca)$ y $Quant_E^{-1}(pLevel)$ se utilizan para valores residuales de predicción en el modo de salto de transformada, donde $pEsca$ se reemplazaría con los valores residuales

de predicción $pResi$, y la salida de $Quant_E^{-1}(pLevel)$ en este caso sería los valores residuales de predicción reconstruidos $pResi'$.

Específicamente, la operación de cuantificación se describe como:

$$pLevel = pResi \times \frac{encScale[QP\%6]}{2^{14} \times 2^{(QP/6)}}$$

donde $pLevel$ es el nivel cuantificado, $pResi$ es el valor de muestra residual de predicción, $encScale[]$ es el valor de escala, QP es el parámetro de cuantificación, $\%$ indica la operación de módulo, $QP\%6$ representa la operación del módulo QP 6.

5 Además, la operación de descuantificación se describe como:

$$pResi' = pLevel \times \frac{decScale[QP\%6] \times 2^{(QP/6)}}{2^6}$$

donde $pLevel$ es el nivel cuantificado, $pResi'$ es el valor de muestra residual de predicción reconstruida, $decScale[]$ es el valor de escala, QP es el parámetro de cuantificación, $\%$ denota la operación de módulo, $QP\%6$ representa la operación del módulo QP 6.

10 Armonización de la cuantificación y descuantificación utilizadas para el modo de transformada regular, el modo de salto de transformada y el modo de paleta

En algunos ejemplos, las operaciones de cuantificación y descuantificación del modo de transformada regular, el modo de salto de transformada y el modo de paleta se unifican todas aplicando las operaciones de cuantificación y descuantificación para el modo de transformada regular a todos los modos, incluyendo también el modo de salto de transformada y el modo de paleta.

15 En algunos ejemplos, los procesos de cuantificación y descuantificación para el modo de transformada regular se utilizan también para el modo de salto de transformada y el modo de paleta. Por ejemplo, las ecuaciones (2), (3), (4) y (5) también se utilizan para el proceso de cuantificación/descuantificación en modo de salto de transformada y modo de paleta.

En algunos ejemplos, cuando las funciones $Quant(pCoeff)$ y $Quant^{-1}(pLevel)$ se utilizan para valores residuales de predicción $pResi$ en el modo de salto de transformada, $pCoeff$ sería reemplazada con $pResi \cdot 2^{transformShift}$ y la salida de $Quant^{-1}(pLevel)$ en este caso sería igual a $pResi' \cdot 2^{transformShift}$. Como resultado, el valor residual de predicción reconstruido $pResi'$ se deriva por $Quant^{-1}(pLevel) / 2^{transformShift}$. Por consiguiente, excepto algunas operaciones de desplazamiento, las mismas funciones de cuantificación y descuantificación para coeficientes de salto de transformada se utilizan para valores residuales de predicción bajo modo de salto de transformada.

20 En algunos ejemplos, cuando las funciones $Quant(pCoeff)$ y $Quant^{-1}(pLevel)$ se utilizan para valores residuales de predicción $pResi$ en el modo de salto de transformada, $pCoeff$ sería reemplazada con $pResi \cdot 2^{transformShift}$ y la salida de $Quant^{-1}(pLevel)$ en este caso sería igual a $pResi' \cdot 2^{transformShift}$. Como resultado, el valor residual de predicción reconstruido $pResi'$ se deriva por $Quant^{-1}(pLevel) / 2^{transformShift}$. Por consiguiente, excepto algunas operaciones de desplazamiento, las mismas funciones de cuantificación y descuantificación para coeficientes de salto de transformada se utilizan para valores residuales de predicción bajo modo de salto de transformada.

25

Específicamente, la operación de cuantificación puede describirse como:

$$pLevel = pResi \times 2^{transformShift} \times \frac{encScale[rectNonTsFlag][QP\%6]}{2^{14} \times 2^{(QP/6)} \times 2^{transformShift'}}$$

$$transformShift = 15 - bitDepth - (\log_2(W) + \log_2(H))/2$$

$$transformShift' = rectNonTsFlag? (transformShift - 1): transformShift$$

30 donde $pLevel$ es el nivel cuantificado, $pResi$ es el valor de muestra residual de predicción, $encScale[][]$ es el valor de escala, QP es el parámetro de cuantificación, $\%$ indica la operación de módulo, $QP\%6$ representa la operación del módulo QP 6, W es la anchura de la CU, H es la altura de la CU, $bitDepth$ es la profundidad de bits de codificación, $rectNonTsFlag$ es igual a 0 cuando la CU se clasifica como un bloque normal, y $rectNonTsFlag$ es igual a 1 cuando la CU se clasifica como un bloque compensado.

35 Además, la operación de descuantificación puede describirse como:

$$pResi' \times 2^{transformShift} = pLevel \times \frac{decScale[rectNonTsFlag][QP\%6] \times 2^{(QP/6)} \times 2^{transformShift'}}{2^6}$$

$$transformShift = 15 - bitDepth - (\log_2(W) + \log_2(H))/2,$$

$$transformShift' = rectNonTsFlag? (transformShift - 1): transformShift$$

40 donde $pLevel$ es el nivel cuantificado, $pResi'$ es el valor de muestra residual de predicción reconstruida, $decScale[][]$ es el valor de escala, QP es el parámetro de cuantificación, $\%$ indica la operación de módulo, $QP\%6$ representa la operación del módulo QP 6, W es la anchura de la CU, H es la altura de la CU, $bitDepth$ es la profundidad de bits de codificación, $rectNonTsFlag$ es igual a 0 cuando la CU se clasifica como un bloque normal, y $rectNonTsFlag$ es igual a 1 cuando la CU se clasifica como un bloque compensado.

En algunos ejemplos, cuando las funciones $Quant(pCoeff)$ y $Quant^1(pLevel)$ se utilizan para valores de escape pEsca bajo el modo paleta, pCoeff se reemplazaría con pReca · $2^{transformShift}$ y la salida de $Quant^1(pLevel)$ en este caso sería igual a pEsca' · $2^{transformShift}$. Como resultado, el valor de escape reconstruido pEsca' se deriva por $Quant^1(pLevel)/2^{transformShift}$. Por lo tanto, excepto algunas operaciones de desplazamiento, las mismas funciones de cuantificación y descuantificación para coeficientes de salto de transformada también se utilizan para valores de escape en el modo de paleta.

Específicamente, la operación de cuantificación puede describirse como:

$$pLevel = pEsca \times 2^{transformShift} \times \frac{encScale[rectNonTsFlag][QP\%6]}{2^{14} \times 2^{(QP/6)} \times 2^{transformShift'}}$$

$$transformShift = 15 - bitDepth - (\log_2(W) + \log_2(H))/2$$

$$transformShift' = rectNonTsFlag ? (transformShift - 1) : transformShift$$

en donde pLevel es el nivel cuantificado, pEsca es el valor de color de escape, encScale[[]] es el valor de escala, QP es el parámetro de cuantificación, % indica la operación de módulo, QP%6 representa la operación del módulo QP 6, W es la anchura de la CU, H es la altura de la CU, bitQa es la profundidad de bits de codificación, rectNonTsFlag es igual a 0 cuando la CU se clasifica como un bloque normal, y rectNonTsFlag es igual a 1 cuando la CU se clasifica como un bloque compensado.

Vale la pena señalar que los procesos de cuantificación y descuantificación de acuerdo con $Quant(pCoeff)$ y $Quant^1(pLevel)$ puede no utilizarse para soportar codificación sin pérdidas en VVC porque no se garantiza la reconstrucción perfecta incluso con un tamaño de paso de cuantificación de 1 utilizando estas dos funciones.

En algunos ejemplos, cuando el procedimiento anterior se utiliza para unificar los procesos de cuantificación y descuantificación de diferentes valores de muestra, un indicador separado, por ejemplo, trans_quant_bypass_flag puede señalizarse en el flujo de bits para indicar si un área de imagen dada, por ejemplo, una CU se codifica en modo de codificación sin pérdidas. Si tal indicador indica que un bloque dado está codificado en modo sin pérdidas, los procesos correspondientes de cuantificación y descuantificación son desviados para la codificación del bloque

La FIG. 10 es un diagrama de bloques que ilustra un aparato para la codificación de vídeo de acuerdo con algunas implementaciones de la presente divulgación. El aparato 1100 puede ser un terminal, tal como un teléfono móvil, una tableta, un terminal de difusión digital, un dispositivo de tableta o un asistente digital personal.

Como se muestra en la FIG. 10, el aparato 1000 puede incluir uno o más de los siguientes componentes: un componente 1002 de procesamiento, una memoria 1004, un componente 1006 de suministro de energía, un componente 1008 multimedia, un componente 1010 de audio, una interfaz 1012 de entrada/salida (E/S), un componente 1014 de sensor y un componente 1016 de comunicación.

El componente 1002 de procesamiento controla normalmente las operaciones globales del aparato 1000, tales como las operaciones relacionadas con la visualización, una llamada telefónica, comunicación de datos, una operación de cámara y una operación de grabación. El componente 1002 de procesamiento puede incluir uno o más procesadores 1020 para ejecutar instrucciones para completar todas o una parte de las etapas del procedimiento anterior. Además, el componente 1002 de procesamiento puede incluir uno o más módulos para facilitar la interacción entre el componente 1002 de procesamiento y otros componentes. Por ejemplo, el componente 1002 de procesamiento puede incluir un módulo multimedia para facilitar la interacción entre el componente 1008 multimedia y el componente 1002 de procesamiento.

La memoria 1004 está configurada para almacenar diferentes tipos de datos para soportar operaciones del aparato 1000. Ejemplos de tales datos incluyen instrucciones, datos de contacto, datos de agenda telefónica, mensajes, imágenes, vídeos, etc. para cualquier aplicación o método que opere en el aparato 1000. La memoria 1004 puede implementarse mediante cualquier tipo de dispositivos de almacenamiento volátiles o no volátiles o una combinación de los mismos, y la memoria 1004 puede ser una memoria estática de acceso aleatorio (SRAM), una memoria de solo lectura programable y borrable eléctricamente (EEPROM), una memoria de solo lectura programable y borrable (EPROM), una memoria de solo lectura programable (PROM), una memoria de solo lectura programable (ROM), una memoria magnética, una memoria flash, un disco magnético o un disco compacto.

El componente 1006 de suministro de energía suministra energía para diferentes componentes del aparato 1000. El componente 1006 de suministro de energía puede incluir un sistema de gestión de suministro de energía, uno o más suministros de energía, y otros componentes asociados con la generación, gestión y distribución de energía para el aparato 1000.

El componente 1008 multimedia incluye una pantalla que proporciona una interfaz de salida entre el aparato 1000 y un usuario. En algunos ejemplos, la pantalla puede incluir una pantalla de cristal líquido (LCD) y un panel táctil (TP). Si la pantalla incluye un panel táctil, la pantalla puede implementarse como una pantalla táctil que recibe una señal de

5 entrada de un usuario. El panel táctil puede incluir uno o más sensores táctiles para detectar un toque, un deslizamiento y un gesto en el panel táctil. El sensor táctil puede no solo detectar un límite de una acción de tocar o deslizar, sino también detectar la duración y la presión relacionadas con la operación de tocar o deslizar. En algunos ejemplos, el componente 1008 multimedia puede incluir una cámara delantera y/o una cámara trasera. Cuando el aparato 1000 está en un modo de funcionamiento, tal como un modo de disparo o un modo de vídeo, la cámara delantera y/o la cámara trasera pueden recibir datos multimedia externos.

10 El componente 1010 de audio está configurado para emitir y/o introducir una señal de audio. Por ejemplo, el componente 1010 de audio incluye un micrófono (MIC). Cuando el aparato 1000 está en un modo operativo, tal como un modo de llamada, un modo de grabación y un modo de reconocimiento de voz, el micrófono está configurado para recibir una señal de audio externa. La señal de audio recibida puede además almacenarse en la memoria 1004 o enviarse a través del componente 1016 de comunicación. En algunos ejemplos, el componente 1010 de audio incluye además un altavoz para emitir una señal de audio.

15 La interfaz de E/S 1012 proporciona una interfaz entre el componente de procesamiento 1002 y un módulo de interfaz periférica. El módulo de interfaz periférica anterior puede ser un teclado, una rueda de clic, un botón o similar. Estos botones pueden incluir, pero no se limitan a, un botón de inicio, un botón de volumen, un botón de inicio y un botón de bloqueo.

20 El componente 1014 de sensor incluye uno o más sensores para proporcionar una evaluación de estado en diferentes aspectos para el aparato 1000. Por ejemplo, el componente 1014 de sensor puede detectar un estado de encendido/apagado del aparato 1000 y ubicaciones relativas de componentes. Por ejemplo, los componentes son una pantalla y un teclado del aparato 1000. El componente 1014 de sensor también puede detectar un cambio de posición del aparato 1000 o un componente del aparato 1000, la presencia o ausencia de un contacto de un usuario en el aparato 1000, una orientación o aceleración/desaceleración del aparato 1000 y un cambio de temperatura del aparato 1000. El componente 1014 de sensor puede incluir un sensor de proximidad configurado para detectar la presencia de un objeto cercano sin ningún toque físico. El componente 1014 de sensor puede incluir además un sensor óptico, tal como un sensor de imagen CMOS o CCD utilizado en una aplicación de formación de imágenes. En algunos ejemplos, el componente 1014 de sensor puede incluir además un sensor de aceleración, un sensor de giroscopio, un sensor magnético, un sensor de presión o un sensor de temperatura.

30 El componente de comunicación 1016 está configurado para facilitar la comunicación por cable o inalámbrica entre el aparato 1000 y otros dispositivos. El aparato 1000 puede acceder a una red inalámbrica en base a un estándar de comunicación, tal como WiFi, 4G, o una combinación de los mismos. En un ejemplo, el componente de comunicación 1016 recibe una señal de difusión o información relacionada con la difusión desde un sistema de gestión de difusión externo a través de un canal de difusión. En un ejemplo, el componente de comunicación 1016 puede incluir además un módulo de comunicación de campo cercano (NFC) para promover la comunicación de corto alcance. Por ejemplo, el módulo NFC puede implementarse en base a la tecnología de identificación por radiofrecuencia (RFID), la tecnología de asociación de datos infrarrojos (IrDA), la tecnología de banda ultra ancha (UWB), la tecnología Bluetooth (BT) y otra tecnología.

40 En un ejemplo, el aparato 1000 puede implementarse mediante uno o más de circuitos integrados de aplicación específica (ASIC), procesadores de señales digitales (DSP), dispositivos de procesamiento de señales digitales (DSPD), dispositivos lógicos programables (PLD), matrices de puertas programables en campo (FPGA), controladores, microcontroladores, microprocesadores u otros elementos electrónicos para realizar el método anterior.

Un medio de almacenamiento legible por ordenador no transitorio puede ser, por ejemplo, una unidad de disco duro (HDD), una unidad de estado sólido (SSD), memoria flash, una unidad híbrida o una unidad híbrida de estado sólido (SSHD), una memoria de solo lectura (ROM), una memoria de solo lectura de disco compacto (CD-ROM), una cinta magnética, un disquete y etc.

45 La FIG. 11 es un diagrama de flujo que ilustra un proceso ejemplar de diseño de cuantificación en la codificación de vídeo de acuerdo con algunas implementaciones de la presente divulgación. El proceso puede aplicarse en un codificador.

En el paso 1101, el procesador 1020 determina un parámetro de cuantificación para datos residuales de una CU.

50 En el paso 1102, el procesador 1020 deriva un valor de escala mediante escalado del parámetro de cuantificación por un factor de escala.

En el paso 1103, el procesador 1020 determina una pluralidad de coeficientes asociados con la CU.

En algunos ejemplos, la pluralidad de coeficientes puede comprender un coeficiente de transformada, un valor de color de escape para la CU y un valor de muestra residual de predicción.

En el paso 1104, el procesador 1020 determina una pluralidad de parámetros asociados con la CU.

En algunos ejemplos, la pluralidad de parámetros comprende un desplazamiento de bits que se determina en base a una profundidad de bits de codificación, y una anchura y una altura de la CU, y el factor de escala se determina en base a la anchura y la altura de la CU.

5 En el paso 1105, el procesador 1020 obtiene una pluralidad de desplazamientos de bits mediante el desplazamiento de bits la pluralidad de parámetros.

En el paso 1106, el procesador 1020 obtiene un nivel cuantificado en base al valor de escala, la pluralidad de coeficientes y la pluralidad de desplazamientos de bits.

10 En algunos ejemplos, cuando el procesador 1020 determina la pluralidad de coeficientes asociados con la CU, el procesador 1020 determina además un valor de color de escape para la CU. El valor de color de escape puede ser un valor para un píxel en la CU con un color que no está en una pluralidad preestablecida de colores seleccionados de la CU. Y cuando el procesador 1020 obtiene el nivel cuantificado en base al valor de escala, la pluralidad de coeficientes y la pluralidad de desplazamientos de bits, el procesador 1020 obtiene además el nivel cuantificado en base al valor de escala, el valor de color de escape y la pluralidad de desplazamientos de bits.

15 En algunos ejemplos, el procesador 1020 determina además una muestra residual de predicción asociada con la CU. Si el procesador 1020 determina la pluralidad de coeficientes asociados con la CU, el procesador 1020 determina además un valor de muestra residual de predicción correspondiente a la muestra residual de predicción. Cuando el procesador 1020 obtiene el nivel cuantificado en base al valor de escala, la pluralidad de coeficientes y la pluralidad de desplazamientos de bits, el procesador 1020 obtiene además el nivel cuantificado en base al valor de escala, el valor de muestra residual de predicción y la pluralidad de desplazamientos de bits.

20 La FIG. 12 es un diagrama de flujo que ilustra un proceso ejemplar de diseño de descuantificación en la codificación de vídeo de acuerdo con algunas implementaciones de la presente divulgación. El proceso puede aplicarse en un decodificador.

En el paso 1201, el procesador 1020 recibe un flujo de bits de vídeo que comprende un parámetro de cuantificación y un nivel cuantificado.

25 En el paso 1202, el procesador 1020 deriva un valor de escala mediante el escalado del parámetro de cuantificación por un factor de escala.

En el paso 1203, el procesador 1020 determina una pluralidad de parámetros asociados con una CU.

30 En algunos ejemplos, la pluralidad de parámetros puede comprender un desplazamiento de bits que se determina en base a una profundidad de bits de codificación, y una anchura y una altura de la CU, y el factor de escala se determina en base al anchura y la altura de la CU.

En el paso 1204, el procesador 1020 obtiene una pluralidad de desplazamientos de bits mediante el desplazamiento de bits la pluralidad de parámetros.

En el paso 1205, el procesador 1020 obtiene una pluralidad de coeficientes asociados con la CU en base al nivel cuantificado, el valor de escala y la pluralidad de desplazamientos de bits.

35 En algunos ejemplos, la pluralidad de coeficientes puede comprender un coeficiente de transformada reconstruido, un valor de color de escape reconstruido para la CU y un valor de muestra residual de predicción reconstruida.

40 En algunos ejemplos, cuando el procesador 1020 obtiene la pluralidad de coeficientes asociados con la CU en base al nivel cuantificado, el valor de escala y la pluralidad de desplazamientos de bits, el procesador 1020 obtiene además un valor de color de escape reconstruido para la CU en base al nivel cuantificado, el valor de escala y la pluralidad de desplazamientos de bits. Y el valor de color de escape reconstruido es un valor para un píxel en la CU con un color que no está en una pluralidad preestablecida de colores seleccionados de la CU.

45 En algunos ejemplos, el procesador 1020 determina además una muestra residual de predicción asociada con la CU. Cuando el procesador 1020 obtiene la pluralidad de coeficientes asociados con la CU en base al nivel cuantificado, el valor de escala y la pluralidad de desplazamientos de bits, el procesador 1020 obtiene el valor de muestra residual de predicción reconstruida para la CU en base al nivel cuantificado, el valor de escala y la pluralidad de desplazamientos de bits. Y el valor de muestra residual de predicción reconstruida puede corresponder a la muestra residual de predicción.

50 En algunos ejemplos, se proporciona un dispositivo informático para la codificación de vídeo. El aparato incluye un procesador 1020; y una memoria 1004 configurada para almacenar instrucciones ejecutables por el procesador; donde el procesador, tras la ejecución de las instrucciones, está configurado para realizar un procedimiento como se ilustra en la FIG. 11.

En algunos ejemplos, se proporciona un dispositivo informático para la codificación de vídeo. El aparato incluye un procesador 1020; y una memoria 1004 configurada para almacenar instrucciones ejecutables por el procesador; donde

el procesador, tras la ejecución de las instrucciones, está configurado para realizar un procedimiento como se ilustra en la FIG. 12.

5 En algunos otros ejemplos, se proporciona un medio 1004 de almacenamiento legible por ordenador no transitorio, que tiene instrucciones almacenadas en el mismo. Cuando las instrucciones son ejecutadas por un procesador 1020, las instrucciones hacen que el procesador realice un método como se ilustra en la FIG. 11.

En algunos otros ejemplos, se proporciona un medio 1004 de almacenamiento legible por ordenador no transitorio, que tiene instrucciones almacenadas en el mismo. Cuando las instrucciones son ejecutadas por un procesador 1020, las instrucciones hacen que el procesador realice un método como se ilustra en la FIG. 12.

10 La descripción de la presente divulgación se ha presentado con fines ilustrativos, y no pretende ser exhaustiva o limitada a la presente divulgación.

REIVINDICACIONES

1. Procedimiento para la descodificación de vídeo, que comprende:

derivar, a partir de un flujo de bits y para un primer bloque de codificación, un parámetro de cuantificación y un nivel cuantificado;

5 derivar un valor de escala buscando una tabla de niveles de escala en base al parámetro de cuantificación; y

de acuerdo con una determinación de que el primer bloque de codificación se codifica en un modo de salto de transformada, obtener una muestra residual para una muestra en el primer bloque de codificación en base al nivel cuantificado, el valor de escala y una pluralidad de desplazamientos de bits aplicando una primera operación de descuantificación;

10 en la que la primera operación de descuantificación también se aplica para obtener una muestra reconstruida para una muestra de escape en un segundo bloque de codificación codificado en un modo de paleta; y

en la que un tamaño de bloque del primer bloque de codificación o del segundo bloque de codificación no se utiliza para determinar parámetros de la primera operación de descuantificación,

en la que la primera operación de descuantificación es la siguiente:

15
$$pSample = pLevel \times \frac{decScale[QP\%6] \times 2^{(QP/6)}}{2^6}$$

en donde *pLevel* es un nivel cuantificado, *pSample* corresponde a un valor reconstruido de una muestra residual en el modo de salto de transformada o de una muestra de escape en el modo de paleta, *decScale*[*i*] es un valor de escala, *QP* es un parámetro de cuantificación, % denota una operación de módulo, *QP*%6 representa una operación del módulo *QP* 6.

20 2. El procedimiento de la reivindicación 1, que comprende además:

de acuerdo con una determinación de que un tercer bloque de codificación se codifica en un modo de transformada regular, obtener una muestra residual para una muestra en el tercer bloque de codificación aplicando una segunda operación de descuantificación diferente de la primera operación de descuantificación;

25 en donde el tamaño de bloque del primer bloque de codificación se utiliza para determinar parámetros de la segunda operación de descuantificación.

3. El procedimiento de la reivindicación 1, en donde la tabla de niveles de escala es *levelScale* [*k*] = {40, 45, 51, 57, 64, 72} con *k* = 0, .5.

4. El procedimiento de la reivindicación 1, que comprende además:

30 recibir, en el flujo de bits, un indicador de escape para indicar si una muestra de escape está presente en el segundo bloque de codificación.

5. El procedimiento de la reivindicación 1, en el que la pluralidad de desplazamientos de bits comprende un desplazamiento izquierdo de *QP*/6 bits y un desplazamiento derecho de 6 bits, donde *QP* es el parámetro de cuantificación.

6. Dispositivo informático, que comprende:

35 uno o varios procesadores (1020);

un almacenamiento no transitorio acoplado al uno o más procesadores (1020); y

una pluralidad de programas almacenados en el almacenamiento no transitorio que, cuando son ejecutados por el uno o más procesadores (1020), hacen que el uno o más procesadores (1020) realicen el método de una cualquiera de las reivindicaciones 1-5.

40 7. Un medio de almacenamiento legible por ordenador no transitorio que almacena un flujo de bits para su ejecución por un dispositivo informático que tiene uno o más procesadores (1020), en donde el flujo de bits, cuando se ejecuta por el uno o más procesadores (1020), hace que el uno o más procesadores (1020) realicen el método de una cualquiera de las reivindicaciones 1-5.

45 8. Un producto de programa informático, que comprende instrucciones almacenadas en el mismo, en el que, cuando las instrucciones son ejecutadas por un procesador (1020), las instrucciones hacen que el procesador (1020) realice el método de una cualquiera de las reivindicaciones 1-5.

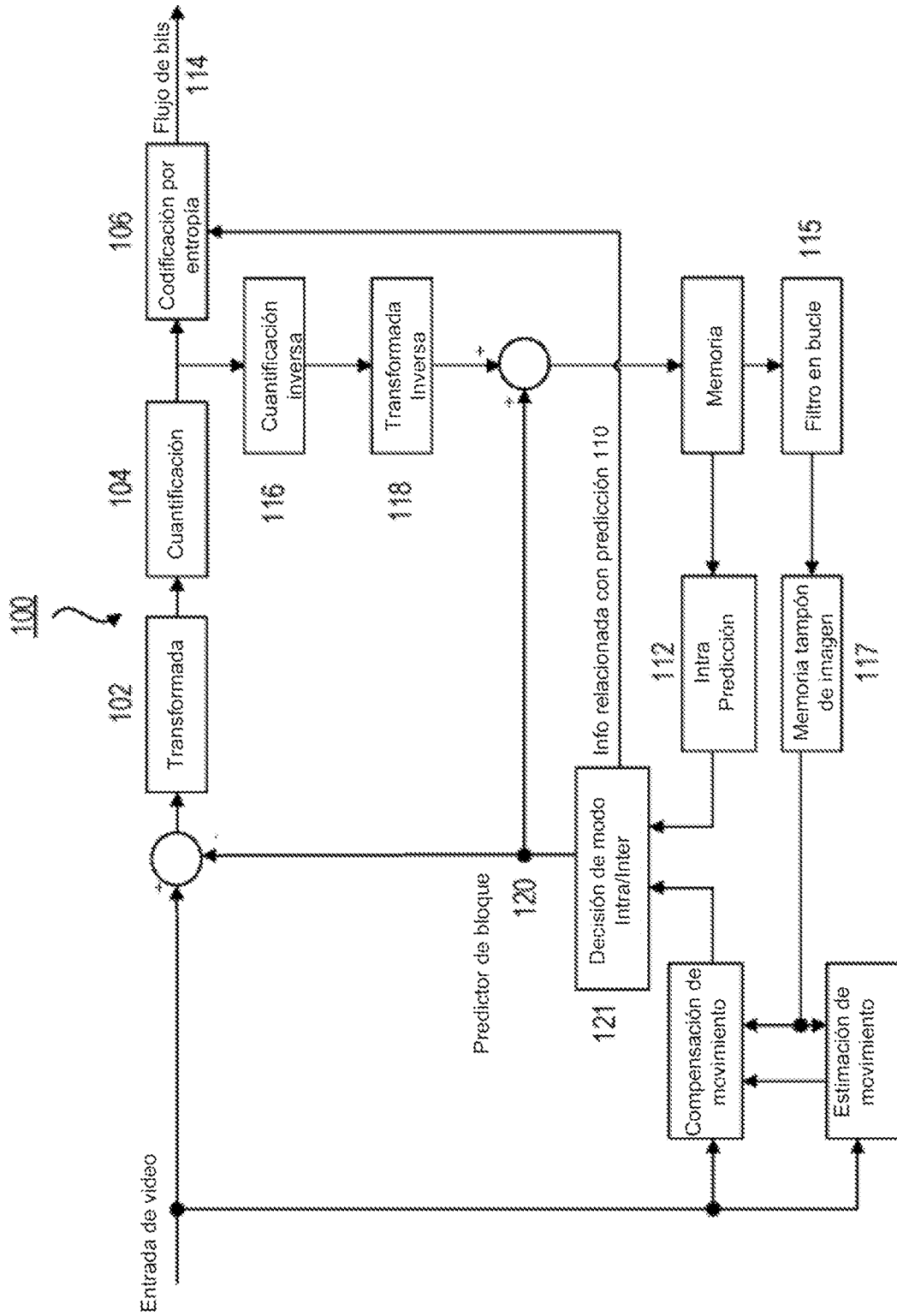


FIG. 1

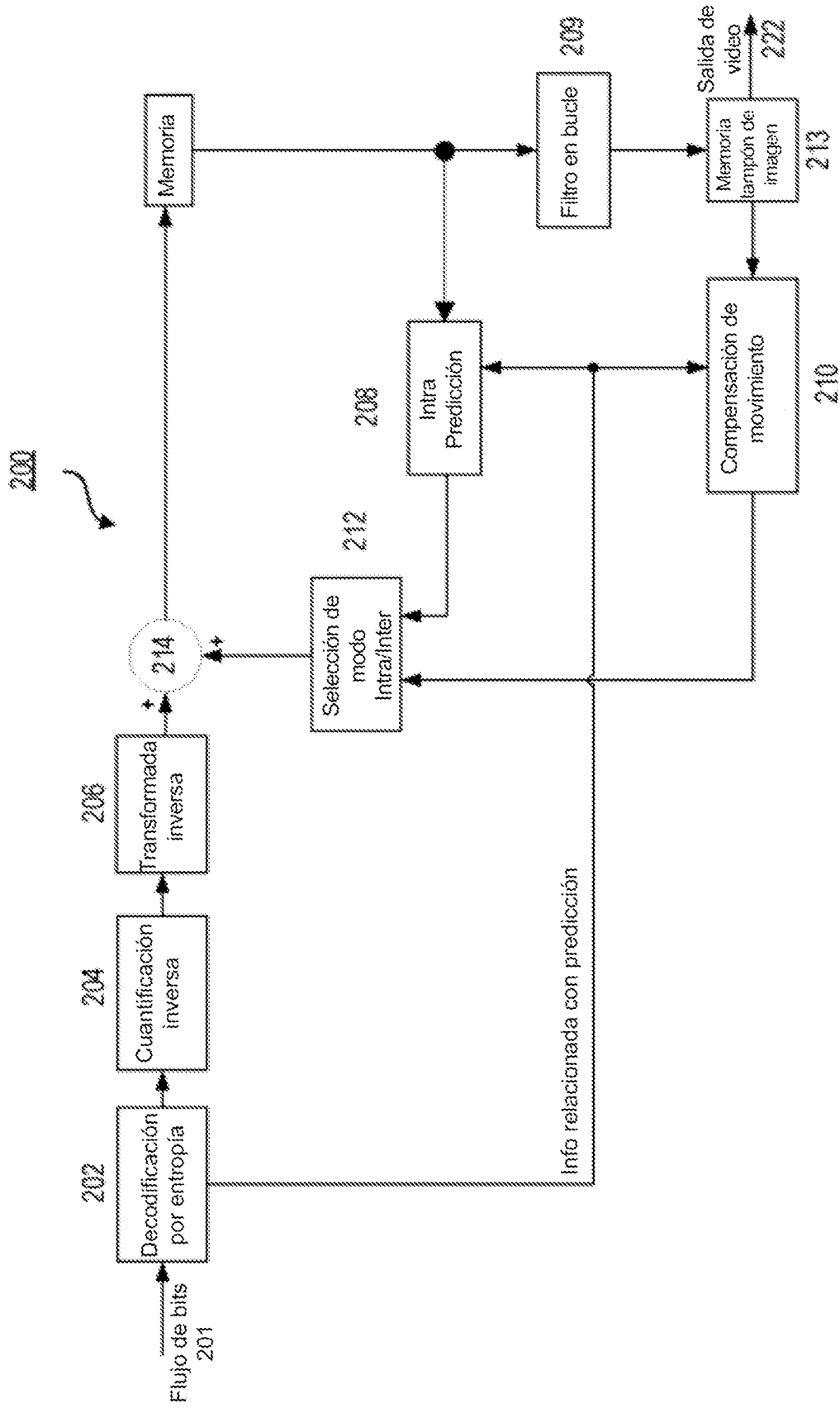


FIG. 2

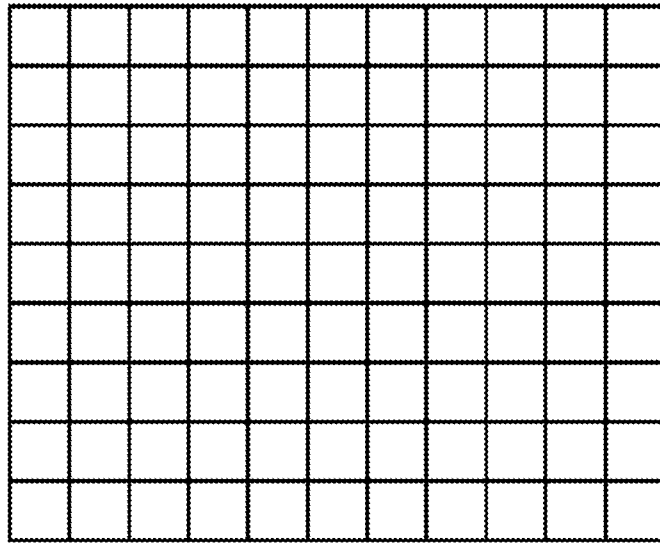
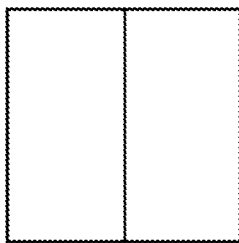
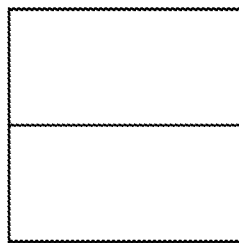


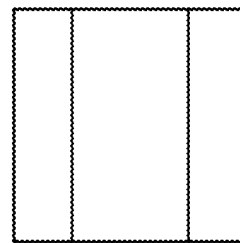
FIG. 3



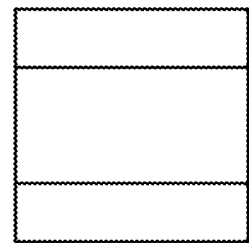
SPLIT_BT_VER



SPLIT_BT_HOR



SPLIT_TT_VER



SPLIT_TT_HOR

FIG. 4

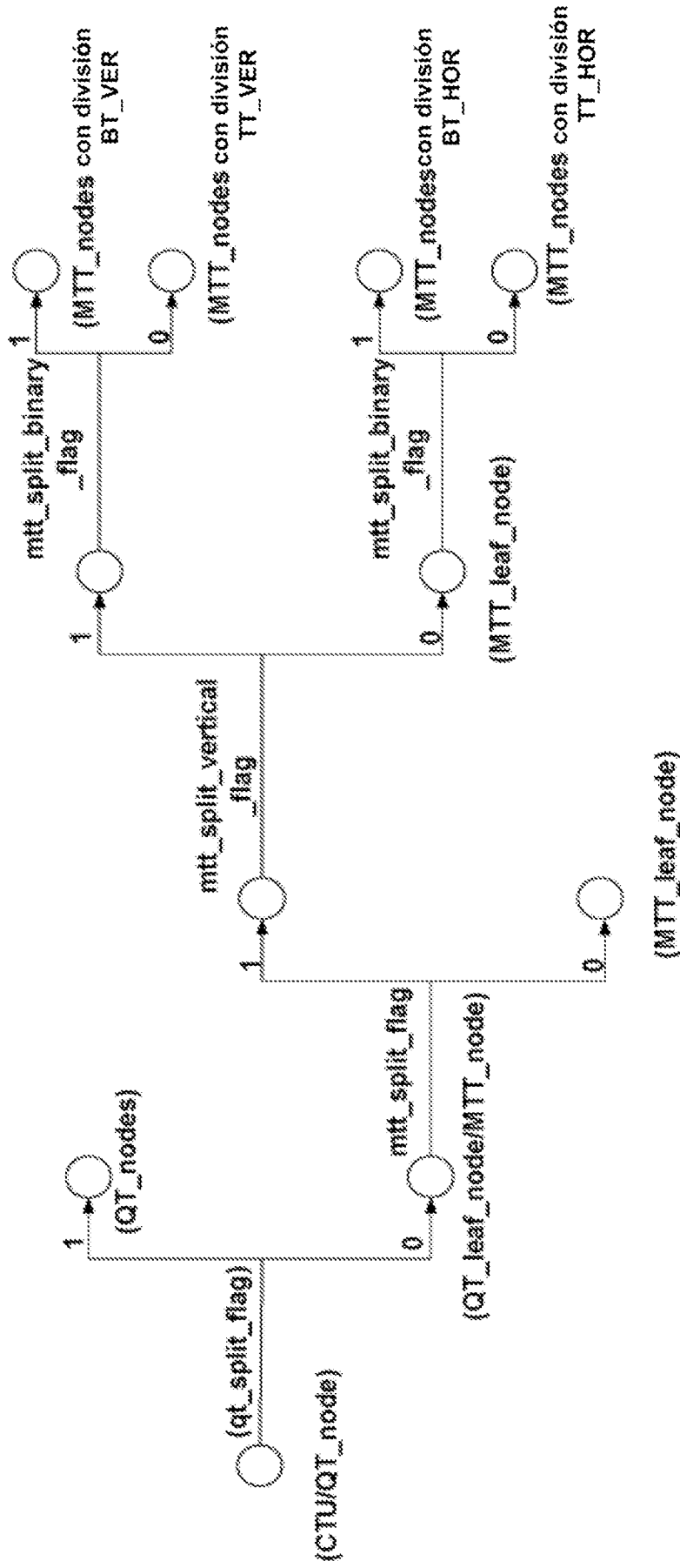


FIG. 5

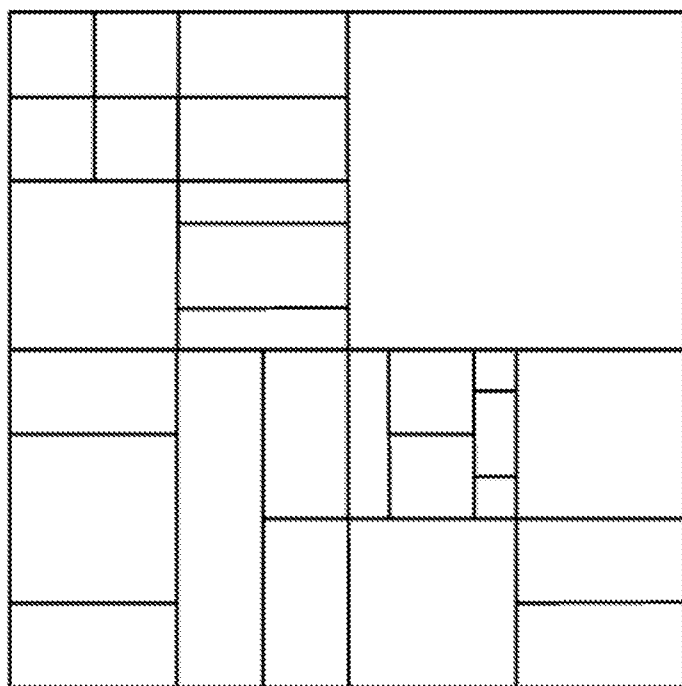


FIG. 6

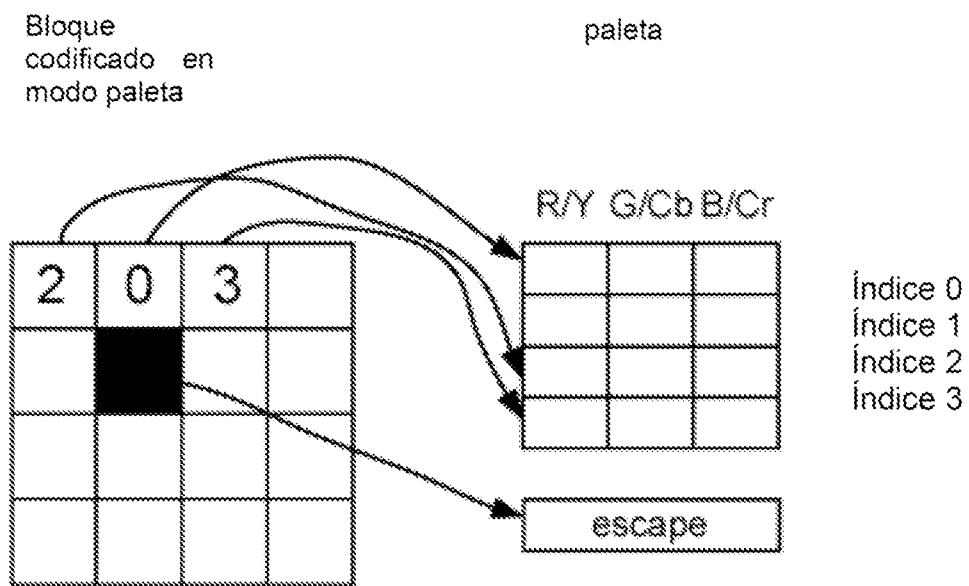


FIG. 7

Exploración
transversal horizontal

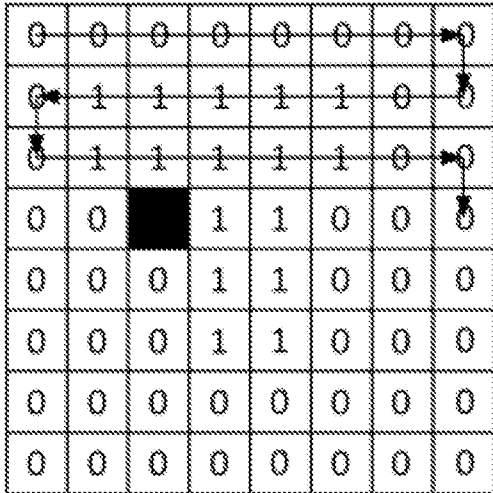


FIG. 8A

Exploración
transversal vertical

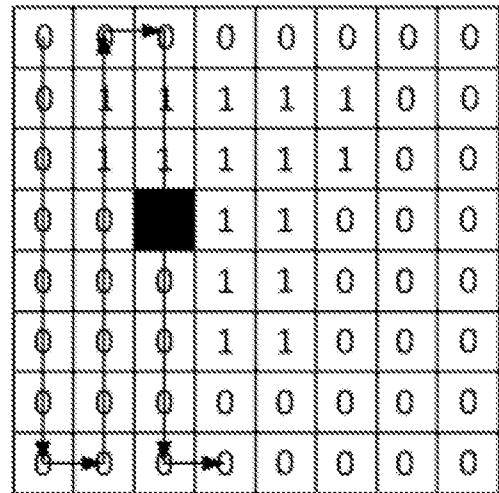


FIG. 8B

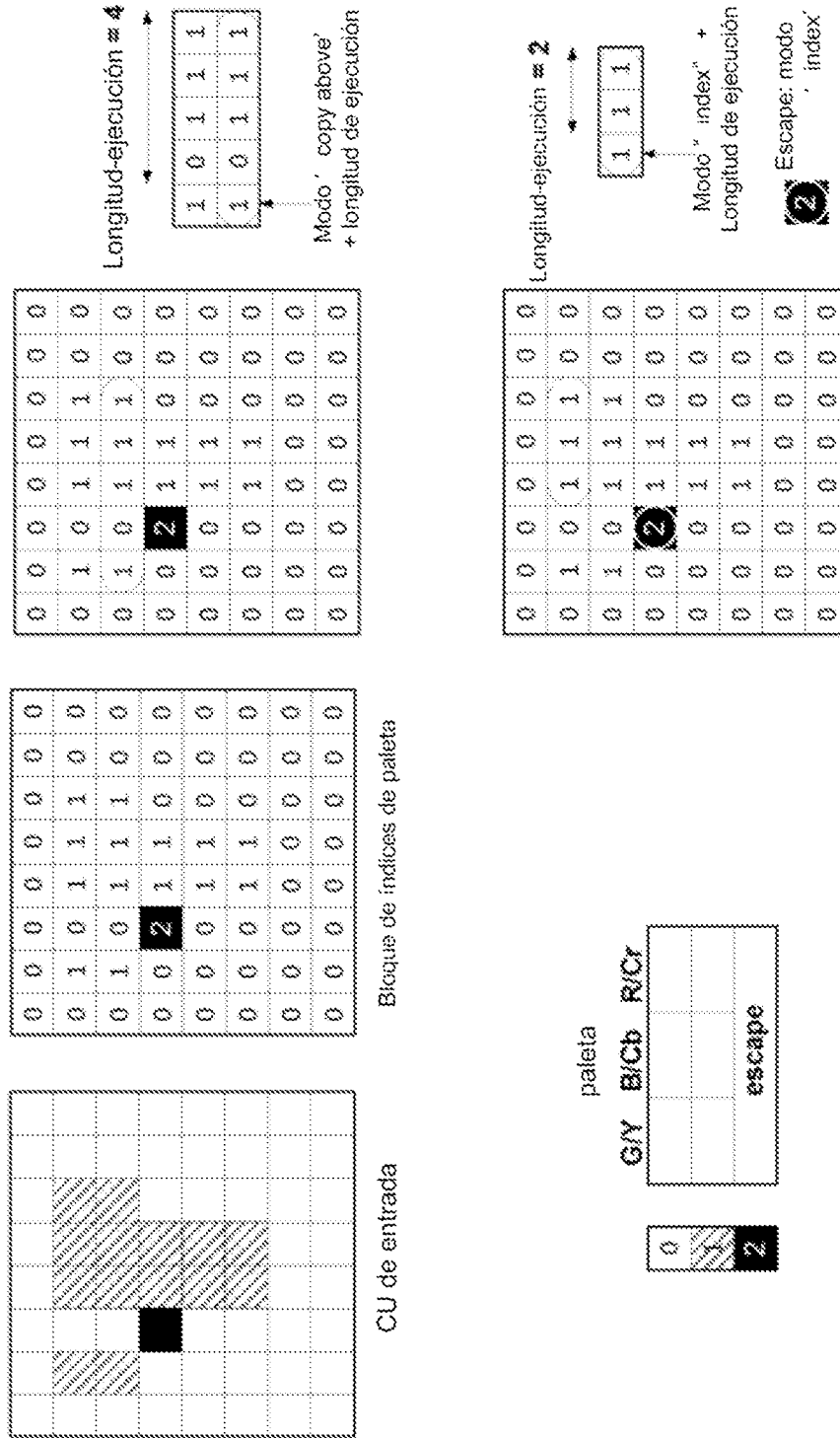


FIG. 9

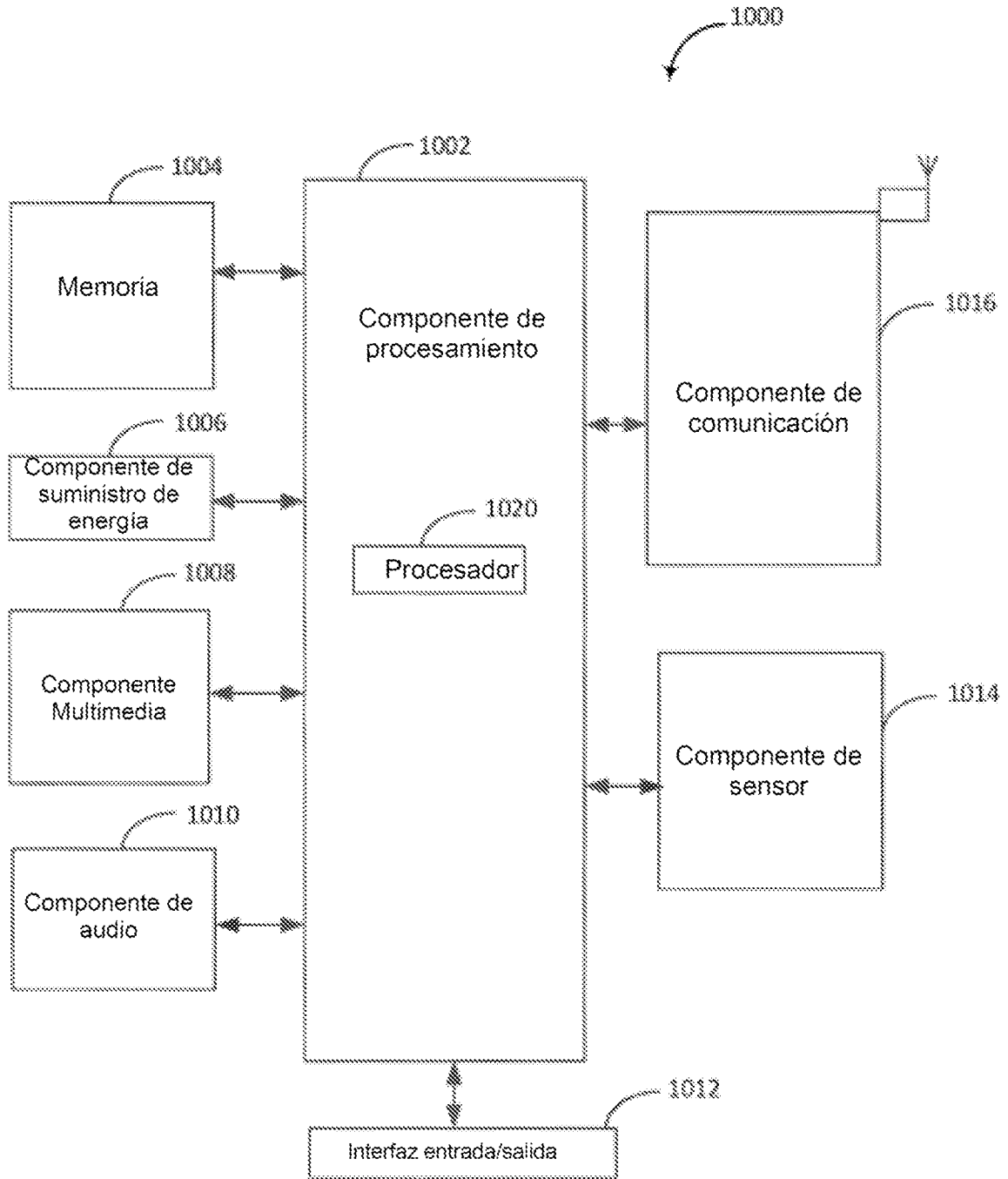
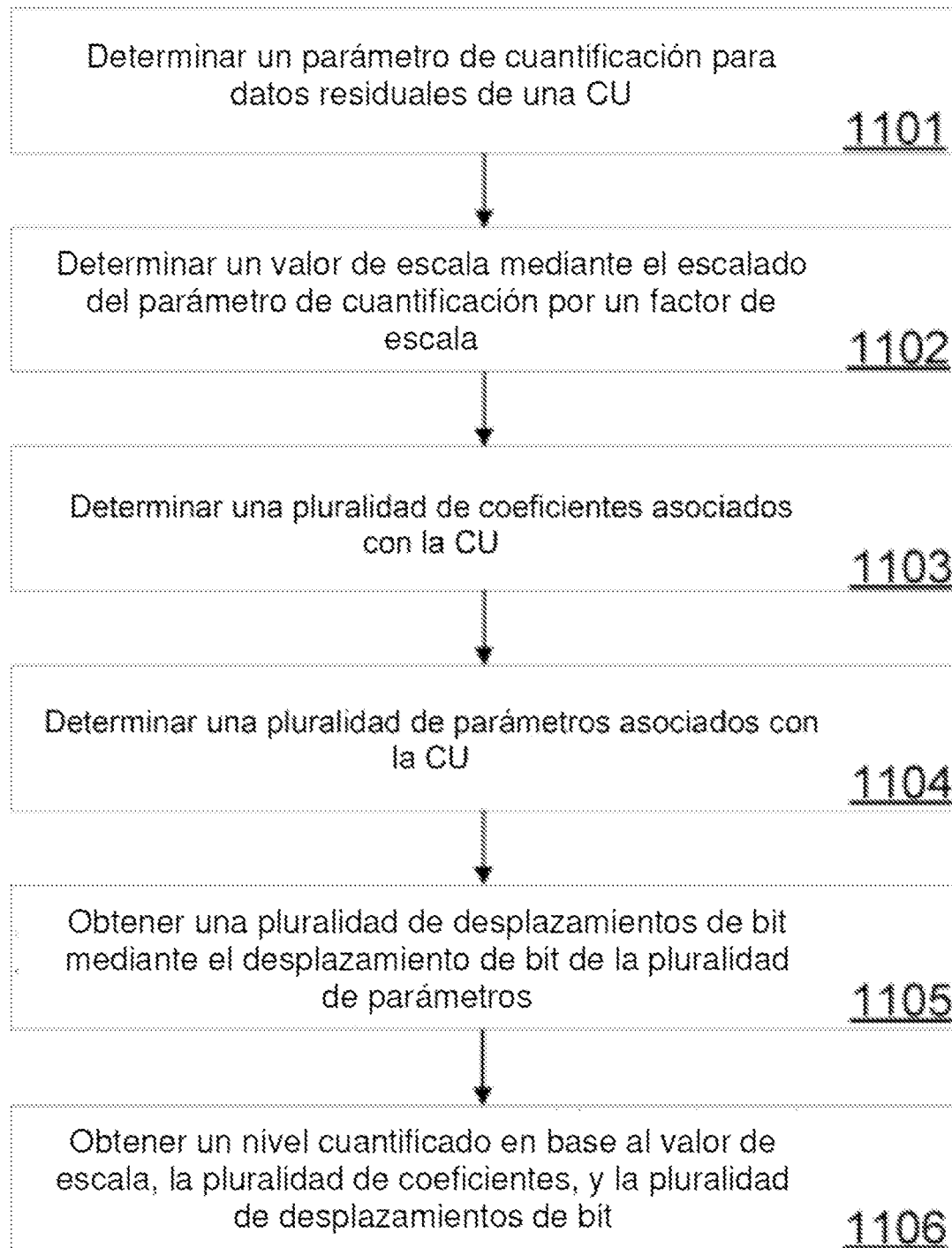
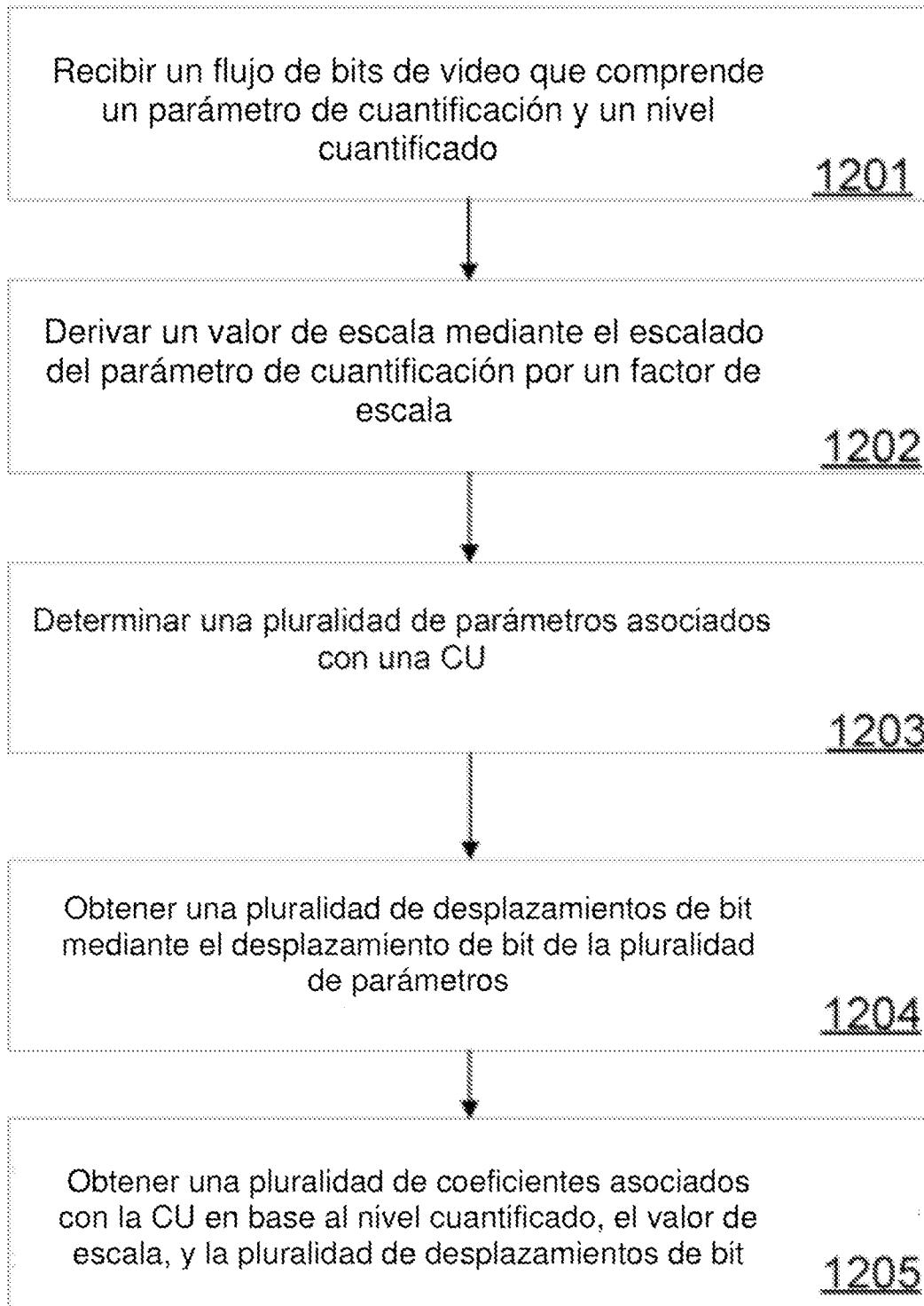


FIG. 10

**FIG. 11**

**FIG. 12**