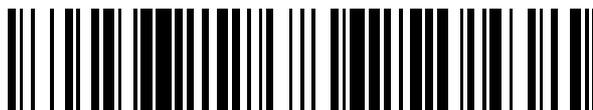


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 770 664**

51 Int. Cl.:

H04N 19/176 (2014.01)

H04N 19/119 (2014.01)

H04N 19/11 (2014.01)

H04N 19/93 (2014.01)

H04N 19/436 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **10.02.2016 PCT/US2016/017258**

87 Fecha y número de publicación internacional: **18.08.2016 WO16130622**

96 Fecha de presentación y número de la solicitud europea: **10.02.2016 E 16706487 (2)**

97 Fecha y número de publicación de la concesión europea: **06.11.2019 EP 3257246**

54 Título: **Restricción en el tamaño de bloque de paleta en la codificación de vídeo**

30 Prioridad:

10.02.2015 US 201562114537 P
09.02.2016 US 201615019086

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
02.07.2020

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US

72 Inventor/es:

SEREGIN, VADIM;
JOSHI, RAJAN LAXMAN;
PU, WEI y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 770 664 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

Restricción en el tamaño de bloque de paleta en la codificación de vídeo

5 **[0001]** Esta solicitud reivindica el beneficio de la solicitud provisional de patente estadounidense n.º 62/114.537, presentada el 10 de febrero de 2015.

CAMPO TÉCNICO

10 **[0002]** Esta divulgación se refiere a codificar y descodificar contenido y, más concretamente, codificar y descodificar contenido de acuerdo con un modo de codificación basado en paletas.

ANTECEDENTES

15 **[0003]** Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas electrónicas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes",
 20 dispositivos de videoconferencia, dispositivos de emisión de vídeo en continuo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, codificación de vídeo avanzada (AVC), la norma de codificación de vídeo de alta eficacia (HEVC) actualmente en desarrollo y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital con
 25 más eficacia implementando dichas técnicas de compresión de vídeo.

[0004] Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (entre imágenes) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (es decir, una trama de vídeo o una porción de una trama de vídeo) se puede dividir en bloques de vídeo. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo de un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma imagen, o predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.
 30
 35

[0005] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y de acuerdo con datos residuales. Para una mayor compresión, los datos residuales se pueden transformar desde el dominio del píxel hasta un dominio de transformada, dando como resultado coeficientes residuales, que a continuación se pueden cuantificar. Los coeficientes cuantificados, dispuestos inicialmente en una matriz bidimensional, se pueden explorar para generar un vector unidimensional de coeficientes, y se puede aplicar codificación por entropía para lograr incluso más compresión.
 40
 45

[0006] El contenido, por ejemplo una imagen, puede codificarse y descodificarse utilizando el modo de paleta. En general, el modo de paleta es una técnica que implica el uso de una paleta de valores de color para representar el contenido. El contenido puede codificarse de manera que el contenido esté representado por un mapa de índices que incluye valores correspondientes a los valores de color en la paleta. El mapa de índices puede descodificarse para obtener valores de color para reconstruir el contenido.
 50

[0007] El estado de la técnica y el problema técnico abordado por la solicitud actual se exponen en la parte central de la descripción, al divulgar la invención.
 55

SUMARIO

[0008] Las técnicas de esta divulgación se refieren a la codificación de contenido basada en paletas. Por ejemplo, en la codificación de contenido basada en paletas, un codificador de contenido (un codificador de contenido tal como un codificador de vídeo o un descodificador de vídeo) puede formar una "paleta" como una tabla de colores para representar los datos de vídeo del área en particular (por ejemplo, un bloque dado). La codificación de contenido basada en paletas puede ser especialmente útil para codificar áreas de datos de vídeo que tienen un número de colores relativamente pequeño. En lugar de codificar valores de píxeles reales (o sus residuales), el codificador de vídeo puede codificar índices de paletas (por ejemplo, valores de índice) para uno o más de los píxeles que relacionan los píxeles con entradas en la paleta que representa los colores de los píxeles. Las técnicas descritas en
 60
 65

esta divulgación pueden incluir técnicas para diversas combinaciones de uno o más entre señalización de modos de codificación basados en paletas, transmisión de paletas, derivación de paletas, derivación del valor de elementos de sintaxis no transmitidos, transmisión de mapas de codificación basados en paletas y otros elementos de sintaxis, predicción de entradas de paleta, codificación de ejecuciones de índices de paleta, información de codificación por entropía de paletas y diversas otras técnicas de codificación de paletas.

[0009] La invención se define por las reivindicaciones independientes adjuntas.

[0010] Los detalles de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetivos y ventajas de la divulgación serán evidentes a partir de la descripción y dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0011]

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo de ejemplo que puede utilizar las técnicas descritas en esta divulgación.

La FIG. 2 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede realizar las técnicas descritas en esta divulgación.

La FIG. 3 es un diagrama de bloques que ilustra un descodificador de vídeo de ejemplo que puede realizar las técnicas descritas en esta divulgación.

La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de determinación de entradas de paleta para la codificación de vídeo basada en paletas, congruente con las técnicas de esta divulgación.

La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de determinación de índices en una paleta para un bloque de píxeles, congruente con las técnicas de esta divulgación.

La FIG. 6 es un diagrama conceptual que ilustra un ejemplo de determinación de una longitud de ejecución máxima con copia arriba, suponiendo un orden de barrido de trama, congruente con las técnicas de esta divulgación.

La FIG. 7 es un diagrama de flujo que ilustra un proceso de ejemplo para procesar datos de vídeo, congruente con las técnicas para la codificación de vídeo basada en paletas de esta divulgación.

DESCRIPCIÓN DETALLADA

[0012] Los aspectos de esta divulgación están dirigidos a técnicas para la codificación de contenido (por ejemplo, codificación de vídeo) y la compresión de datos de contenido (por ejemplo, compresión de datos de vídeo). En particular, esta divulgación describe técnicas para la codificación basada en paletas de datos de contenido (por ejemplo, datos de vídeo). En diversos ejemplos de esta divulgación, las técnicas de esta divulgación pueden dirigirse a procesos de predicción o codificación de un bloque en modo de paleta para mejorar la eficacia de la codificación y/o reducir la complejidad del códec, como se describe más en detalle a continuación. Por ejemplo, la divulgación describe técnicas relacionadas con la restricción del tamaño de bloque de la paleta para el modo de paleta.

[0013] Como se usa en el presente documento, las instancias del término "contenido" pueden cambiarse por el término "vídeo", y las instancias del término "vídeo" pueden cambiarse por el término "contenido". Esto es cierto independientemente de si los términos "contenido" o "vídeo" se están utilizando como adjetivo, sustantivo u otra parte del discurso. Por ejemplo, la referencia a un "codificador de contenido" también incluye una referencia a un "codificador de vídeo", y la referencia a un "codificador de vídeo" también incluye una referencia a un "codificador de contenido". De manera similar, la referencia al "contenido" también incluye la referencia al "vídeo", y la referencia al "vídeo" también incluye la referencia al "contenido".

[0014] Como se usa en el presente documento, "contenido" se refiere a cualquier tipo de contenido. Por ejemplo, "contenido" puede referirse a vídeo, contenido de pantalla, imagen, cualquier contenido gráfico, cualquier contenido visualizable o cualquier dato correspondiente al mismo (por ejemplo, datos de vídeo, datos de contenido de pantalla, datos de imágenes, datos de contenido gráfico, datos de contenido visualizable y similares).

[0015] Como se usa en el presente documento, el término "vídeo" puede referirse al contenido de pantalla, contenido móvil, una pluralidad de imágenes que pueden presentarse en una secuencia, o cualquier dato correspondiente al mismo (por ejemplo, datos de contenido de pantalla, datos de contenido móvil, datos de vídeo, datos de imágenes y similares).

[0016] Como se usa en el presente documento, el término "imagen" puede referirse a una sola imagen, una o más imágenes, una o más imágenes entre una pluralidad de imágenes correspondientes a un vídeo, una o más imágenes entre una pluralidad de imágenes que no corresponden a un vídeo, una pluralidad de imágenes correspondientes a un vídeo (por ejemplo, todas las imágenes correspondientes al vídeo o menos que todas las imágenes correspondientes al vídeo), una subparte de una única imagen, una pluralidad de subpartes de una única imagen, una pluralidad de subpartes correspondientes a una pluralidad de imágenes, una o más primitivas gráficas, datos de imágenes, datos gráficos y similares.

[0017] En la codificación de vídeo tradicional, se supone que las imágenes son de tono continuo y espacialmente homogéneas. En base a estos supuestos, se han desarrollado diversas herramientas, tales como transformadas basadas en bloques, el filtrado y otras herramientas de codificación, y dichas herramientas han demostrado unos buenos resultados para los vídeos de contenido natural. Sin embargo, en aplicaciones como las de escritorio remoto, trabajo colaborativo y visualización inalámbrica, el contenido de pantalla generado por ordenador puede ser el contenido dominante que se debe comprimir. Este tipo de contenido de pantalla tiende a tener un tono discreto, líneas nítidas y límites de objetos de alto contraste. El supuesto de tono continuo y homogeneidad puede no valer ya y, por lo tanto, las técnicas tradicionales de codificación de vídeo pueden ser ineficaces en la compresión de contenido (por ejemplo, contenido de pantalla).

[0018] En un ejemplo de codificación de vídeo basado en paletas, un codificador de vídeo puede codificar un bloque de datos de vídeo determinando una paleta para el bloque (por ejemplo, codificando la paleta explícitamente, prediciendo la paleta, o una combinación de los mismos), ubicando una entrada en la paleta para representar el valor de uno o más píxeles, y codificando tanto la paleta como el bloque con valores de índices que indican la entrada en la paleta utilizada para representar los valores de píxeles del bloque. En algunos ejemplos, el codificador de vídeo puede señalar la paleta y/o los valores de índices en un flujo de bits codificado. A su vez, el descodificador de vídeo puede obtener, a partir de un flujo de bits codificado, una paleta para un bloque, así como valores de índices para los píxeles individuales del bloque. El descodificador de vídeo puede relacionar los valores de índices de los píxeles con las entradas de la paleta para reconstruir los diversos valores de píxeles del bloque.

[0019] Por ejemplo, se puede suponer que un área en particular de datos de vídeo tiene un número de colores relativamente pequeño. Un codificador de vídeo (por ejemplo, un codificador de vídeo o descodificador de vídeo) puede codificar (por ejemplo, codificar o descodificar) una llamada "paleta" para representar los datos de vídeo del área en particular. La paleta puede expresarse como un índice (por ejemplo, tabla) de colores o valores de píxeles que representan los datos de vídeo del área en particular (por ejemplo, un bloque dado). El codificador de vídeo puede codificar el índice, que relaciona uno o más valores de píxeles con el valor apropiado en la paleta. Cada píxel se puede asociar a una entrada en la paleta que representa el color del píxel. Por ejemplo, la paleta puede incluir los valores de píxeles más dominantes en el bloque dado. En algunos casos, los valores de píxeles más dominantes pueden incluir los uno o más valores de píxeles que se producen con mayor frecuencia dentro del bloque. Además, en algunos casos, un codificador de vídeo puede aplicar un valor umbral para determinar si un valor de píxel se incluirá como uno de los valores de píxeles más dominantes en el bloque. De acuerdo con diversos aspectos de la codificación basada en paletas, el codificador de vídeo puede codificar valores de índices indicativos de uno o más de los valores de píxeles del bloque actual, en lugar de codificar valores de píxeles reales o sus residuales para un bloque actual de datos de vídeo. En el contexto de la codificación basada en paletas, los valores de índices indican entradas respectivas en la paleta que se usan para representar valores de píxeles individuales del bloque actual. El ejemplo anterior pretende proporcionar una descripción general de la codificación de vídeo basada en paletas.

[0020] La codificación basada en paletas, que puede ser particularmente adecuada para la codificación de contenido generado por pantalla u otro contenido en el que una o más herramientas de codificación tradicionales son ineficaces. Las técnicas para la codificación basada en paletas de datos de vídeo se pueden usar con una o más técnicas de codificación, tales como técnicas de codificación interpredictiva o intrapredictiva. Por ejemplo, como se describe más en detalle a continuación, un codificador o descodificador, o un codificador-descodificador combinado (códec), puede estar configurado para realizar una codificación interpredictiva e intrapredictiva, así como una codificación basada en paletas.

[0021] En algunos ejemplos, las técnicas de codificación basada en paletas pueden estar configuradas para su uso con una o más normas de codificación de vídeo. Por ejemplo, la Codificación de Vídeo de Alta Eficacia (HEVC) es una nueva norma de codificación de vídeo desarrollada por el Equipo Conjunto de Colaboración en Codificación de vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo del UIT-T (VCEG) y el Grupo de Expertos en Imagen en Movimiento de ISO/IEC (MPEG). El documento estándar finalizado de la HEVC se publica como "UIT-T H.265, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services - Coding of moving video - High efficiency video coding", Sector de normalización de las telecomunicaciones de la Unión Internacional de Telecomunicaciones (ITU), abril de 2013.

[0022] Para proporcionar una codificación más eficaz del contenido generado por la pantalla, el JCT-VC está desarrollando una extensión de la norma HEVC, denominada la norma de codificación de contenido de pantalla (SCC) de la HEVC. Un borrador de trabajo reciente de la norma SCC de la HEVC, denominado "HEVC SCC Draft 2" o "WD2", se describe en el documento JCTVC-S1005, R. Joshi y J. Xu, "HEVC screen content coding draft text 2",

Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 19.ª reunión: Estrasburgo, FR, 17-24 de octubre de 2014.

5 **[0023]** Con respecto al marco de trabajo de la HEVC, como ejemplo, las técnicas de codificación basada en paletas pueden estar configuradas para su uso como un modo de unidad de codificación (CU). En otros ejemplos, las técnicas de codificación basada en paletas pueden estar configuradas para su uso como un modo de unidad de predicción (PU) en el marco de la HEVC. Por consiguiente, todos los procesos siguientes divulgados, descritos en el contexto de un modo de CU, se pueden aplicar, de forma adicional o alternativa, a una PU. Sin embargo, estos ejemplos basados en la HEVC no se deben considerar como una restricción o limitación de las técnicas de codificación basada en paletas descritas en el presente documento, ya que dichas técnicas se pueden aplicar para que funcionen independientemente o como parte de otros sistemas/normas existentes o todavía por desarrollar. En estos casos, la unidad para la codificación de paletas puede ser bloques cuadrados, bloques rectangulares o incluso regiones de forma no rectangular.

15 **[0024]** En algunos ejemplos, se puede derivar una paleta para una o más CU, PU o cualquier región de datos (por ejemplo, cualquier bloque de datos). Por ejemplo, una paleta puede comprender (y puede consistir en) los valores de píxeles más dominantes en la CU actual, donde una CU es la región de datos para este ejemplo en particular. El tamaño y los elementos de la paleta primero se transmiten desde un codificador de vídeo a un descodificador de vídeo. El tamaño y/o los elementos de la paleta, para una CU actual que se está codificando, se pueden codificar directamente o de forma predictiva utilizando el tamaño y/o los elementos de la paleta en las CU que son vecinas de la CU actual, es decir, las CU vecinas (por ejemplo, donde las CU vecinas pueden incluir una CU arriba de la CU actual y/o a la izquierda de la CU actual). Después de eso, los valores de píxeles en la CU se codifican en base a la paleta de acuerdo con un cierto orden de barrido. Para cada ubicación de píxeles en la CU, un indicador, por ejemplo, un `palette_flag`, se transmite primero para indicar si el valor del píxel está incluido en la paleta. Para esos valores de píxeles que correlacionan con una entrada en la paleta, el índice de la paleta asociado con esa entrada se señala para la ubicación de píxel dada en la CU. Para aquellos valores de píxeles que no existen en la paleta, se puede asignar un índice especial al píxel y el valor del píxel real se transmite para la ubicación de píxel dada en la CU. Estos píxeles se denominan "píxeles de escape". Un píxel de escape puede codificarse utilizando cualquier procedimiento de codificación por entropía existente, como la codificación de longitud fija, la codificación unaria, etc.

20 **[0025]** Las muestras en un bloque de datos de vídeo pueden procesarse (por ejemplo, barrido) utilizando un orden de barrido de trama horizontal u otro orden de barrido. Por ejemplo, el codificador de vídeo puede convertir un bloque bidimensional de índices de paleta en una matriz unidimensional barriendo los índices de paleta usando un orden de barrido de trama horizontal. Asimismo, el descodificador de vídeo puede reconstruir un bloque de índices de paleta utilizando el orden de barrido de trama horizontal. Por consiguiente, esta divulgación puede referirse a una muestra previa como una muestra que precede a la muestra que se está codificando actualmente en el bloque en el orden de barrido. Se debe tener en cuenta que los barridos que no sean un barrido de trama horizontal, tales como un orden de barrido de trama vertical, también pueden ser aplicables. El ejemplo anterior, así como otros ejemplos expuestos en esta divulgación, están destinados a proporcionar una descripción general de la codificación de vídeo basada en paletas.

35 **[0026]** La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo 10 de ejemplo que puede utilizar las técnicas de esta divulgación. Como se usa en el presente documento, el término "codificador de vídeo" se refiere genéricamente tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" pueden referirse genéricamente a la codificación de vídeo o la descodificación de vídeo. El codificador de vídeo 20 y el descodificador de vídeo 30 del sistema de codificación de vídeo 10 representan ejemplos de dispositivos que pueden estar configurados para realizar técnicas de codificación de vídeo basada en paletas de acuerdo con diversos ejemplos descritos en esta divulgación. Por ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar configurados para codificar de forma selectiva diversos bloques de datos de vídeo, tales como las CU o PU en la codificación HEVC, usando codificación basada en paletas o bien codificación no basada en paletas. Los modos de codificación no basados en paletas pueden referirse a diversos modos de codificación temporal interpredictiva o a modos de codificación espacial intrapredictiva, tales como los diversos modos de codificación especificados por la norma HEVC.

50 **[0027]** Como se muestra en la FIG. 1, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 y un dispositivo de destino 14. El dispositivo de origen 12 genera datos de vídeo codificados. Por consiguiente, el dispositivo de origen 12 se puede denominar dispositivo de codificación de vídeo o aparato de codificación de vídeo. El dispositivo de destino 14 puede descodificar los datos de vídeo codificados generados por el dispositivo de origen 12. Por consiguiente, el dispositivo de destino 14 se puede denominar dispositivo de descodificación de vídeo o aparato de descodificación de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 pueden ser ejemplos de dispositivos de codificación de vídeo o aparatos de codificación de vídeo.

60 **[0028]** El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia gama de dispositivos, incluyendo ordenadores de escritorio, dispositivos informáticos móviles, ordenadores tipo notebook (por ejemplo, portátiles), tabletas electrónicas, descodificadores multimedia, aparatos telefónicos tales como los

denominados teléfonos "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores para vehículos o similares.

5 **[0029]** El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 por medio de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de transferir los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicación que permiten al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de origen 12 puede modular los datos de vídeo codificados de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. Los uno o más medios de comunicación pueden incluir medios de comunicación inalámbrica y/o alámbrica, tales como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. Los uno o más medios de comunicación pueden formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global (por ejemplo, Internet). Los uno o más medios de comunicación pueden incluir encaminadores, conmutadores, estaciones base u otros equipos que facilitan la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

20 **[0030]** En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacena datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento por medio de, por ejemplo, acceso de disco o de acceso de tarjeta. El medio de almacenamiento puede incluir una variedad de medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria flash u otros medios de almacenamiento digital adecuados para almacenar datos de vídeo codificados.

25 **[0031]** En otro ejemplo, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacena los datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder a datos de vídeo codificados almacenados en el servidor de archivos o en otro dispositivo de almacenamiento intermedio mediante emisión en continuo, o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir los datos de vídeo codificados al dispositivo de destino 14. Ejemplos de servidores de archivos incluyen servidores web (por ejemplo, para un sitio web), servidores de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectado a la red (NAS) y unidades de disco local.

35 **[0032]** El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de una conexión de datos estándar, tal como una conexión a Internet. Ejemplos de tipos de conexiones de datos pueden incluir canales inalámbricos (por ejemplo, conexiones Wi-Fi), conexiones alámbricas (por ejemplo, DSL, módem por cable, etc.) o combinaciones de ambos que sean adecuadas para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión en continuo, una transmisión de descarga o una combinación de ambas.

40 **[0033]** El dispositivo de origen 12 y el dispositivo de destino 14 pueden configurarse para realizar una codificación basada en paletas congruente con esta divulgación. No obstante, las técnicas de esta divulgación para la codificación basada en paletas no están limitadas a aplicaciones o a configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como soporte de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, por medio de Internet, codificación de datos de vídeo para su almacenamiento en un medio de almacenamiento de datos, descodificación de datos de vídeo almacenados en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 puede estar configurado para admitir transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la emisión de vídeo en continuo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

55 **[0034]** El sistema de codificación de vídeo 10 ilustrado en la FIG. 1 es simplemente un ejemplo y las técnicas de esta divulgación pueden aplicarse a configuraciones de codificación de vídeo (por ejemplo, codificación de vídeo o descodificación de vídeo) que no incluyan necesariamente ninguna comunicación de datos entre los dispositivos de codificación y descodificación. En otros ejemplos, los datos se recuperan de una memoria local, se emiten en continuo por una red, o similares. Un dispositivo de codificación de vídeo puede codificar y almacenar datos en una memoria, y/o un dispositivo de descodificación de vídeo puede recuperar y descodificar datos de una memoria. En muchos ejemplos, la codificación y la descodificación se realizan mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en una memoria y/o recuperan y descodifican datos de una memoria.

60 **[0035]** En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos ejemplos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. La fuente de vídeo 18 puede incluir un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contenga datos de vídeo previamente capturados, una interfaz de alimentación de vídeo para recibir datos de vídeo desde un proveedor de contenido de vídeo y/o un

sistema de gráficos por ordenador para generar datos de vídeo, o una combinación de dichas fuentes de datos de vídeo.

5 **[0036]** El codificador de vídeo 20 puede codificar datos de vídeo procedentes de la fuente de vídeo 18. En algunos ejemplos, el dispositivo de origen 12 transmite directamente los datos de vídeo codificados al dispositivo de destino 14 por medio de la interfaz de salida 22. En otros ejemplos, los datos de vídeo codificados también pueden almacenarse en un medio de almacenamiento o en un servidor de archivos para su acceso posterior mediante el dispositivo de destino 14 para su descodificación y/o su reproducción.

10 **[0037]** En el ejemplo de la FIG. 1, el dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos ejemplos, la interfaz de entrada 28 incluye un receptor y/o un módem. La interfaz de entrada 28 puede recibir datos de vídeo codificados a través del canal 16. El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En general, el dispositivo de visualización 32 muestra datos de vídeo descodificados. El dispositivo de visualización 15 32 puede comprender una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

20 **[0038]** Esta divulgación puede referirse, en general, al codificador de vídeo 20 que "señaliza" o "transmite" cierta información a otro dispositivo, tal como el descodificador de vídeo 30. El término "señalizar" o "transmitir" puede referirse en general a la comunicación de elementos de sintaxis y/u otros datos usados para descodificar los datos de vídeo comprimidos. Dicha comunicación puede producirse en tiempo real o casi real. De forma alternativa, dicha comunicación puede producirse durante un tramo de tiempo, tal como podría producirse cuando se almacenan 25 elementos de sintaxis en un medio de almacenamiento legible por ordenador en un flujo de bits codificado en el momento de la codificación, que pueden ser recuperados después por un dispositivo de descodificación en cualquier momento tras haber sido almacenados en este medio. Por lo tanto, aunque el descodificador de vídeo 30 puede denominarse como "que recibe" cierta información, la información que recibe no necesariamente se produce en tiempo real o casi real, y puede recuperarse desde un medio en algún momento después del almacenamiento.

30 **[0039]** Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden implementarse como cualquiera de una variedad de circuitería adecuada, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, hardware o cualquier combinación de los mismos. Si las técnicas se implementan 35 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 divulgación. Cualquiera de lo anterior (incluyendo hardware, software, una combinación de hardware y software, etc.) puede considerarse como uno o más procesadores. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden incluir en uno o más 40 codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

45 **[0040]** En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 funcionan de acuerdo con una norma de compresión de vídeo, tal como la norma HEVC mencionada anteriormente y descrita en la norma HEVC. Además de la norma HEVC base, hay esfuerzos constantes para producir ampliaciones de codificación de vídeo escalable, codificación de vídeo multivista y codificación en 3D para la HEVC. Además, pueden proporcionarse modos de codificación basada en paletas, por ejemplo, como se describe en esta divulgación, para la ampliación de la norma HEVC. En algunos ejemplos, las técnicas descritas en esta divulgación para la 50 codificación basada en paletas se pueden aplicar a codificadores y descodificadores configurados para funcionar de acuerdo con otras normas de codificación de vídeo. Por consiguiente, la aplicación de un modo de codificación basada en paletas para la codificación de unidades de codificación (CU) o unidades de predicción (PU) en un códec de la HEVC se describe con fines de ejemplo.

55 **[0041]** En la HEVC y otras normas de codificación de vídeo, una secuencia de vídeo incluye típicamente una serie de imágenes. Las imágenes también pueden denominarse "tramas". Una imagen puede incluir tres matrices de muestras, indicadas con S_L , S_{Cb} y S_{Cr} . S_L es una matriz bidimensional (es decir, un bloque) de muestras de luma. S_{Cb} es una matriz bidimensional de muestras de crominancia Cb. S_{Cr} es una matriz bidimensional de muestras de crominancia Cr. Las muestras de crominancia también se pueden denominar en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede incluir solo una matriz de muestras de 60 luma.

65 **[0042]** Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede ser un bloque de árbol de codificación de muestras de luma, dos bloques de árbol de codificación correspondientes de muestras de croma y estructuras de sintaxis usadas para codificar las muestras de los bloques de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque de muestras $N \times N$. Una CTU también puede denominarse "bloque de árbol" o "unidad de codificación de máximo tamaño" (LCU). Las CTU de la HEVC pueden ser análogas en términos

generales a los macrobloques de otras normas, tales como H.264/AVC. Sin embargo, una CTU no está necesariamente limitada a un tamaño en particular y puede incluir una o más unidades de codificación (CU). Un fragmento puede incluir un número entero de CTU ordenadas consecutivamente en el barrido de trama. Un fragmento codificado puede comprender una cabecera de fragmento y datos de fragmento. La cabecera de fragmento de un fragmento puede ser una estructura de sintaxis que incluye elementos de sintaxis que proporcionan información sobre el fragmento. Los datos del fragmento pueden incluir CTU codificadas del fragmento.

[0043] Esta divulgación puede usar el término "unidad de vídeo", "bloque de vídeo" o "bloque" para referirse a uno o más bloques de muestras y estructuras de sintaxis usadas para codificar muestras de los uno o más bloques de muestras. Los tipos de ejemplo de unidades o bloques de vídeo pueden incluir CTU, CU, PU, unidades de transformadas (TU), macrobloques, particiones de macrobloque, etc. En el ejemplo de la HEVC, para generar una CTU codificada, el codificador de vídeo 20 puede realizar de forma recursiva una partición de árbol cuaternario en los bloques de árbol de codificación de una CTU para dividir los bloques de árbol de codificación en bloques de codificación, de ahí el nombre "unidades de árbol de codificación". Un bloque de codificación es un bloque de muestras NxN. Una CU puede ser un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene una matriz de muestras de luma, una matriz de muestras de Cb y una matriz de muestras de Cr, y estructuras de sintaxis usadas para codificar las muestras de los bloques de codificación. El codificador de vídeo 20 puede dividir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción puede ser un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las que se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede ser un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma de una imagen y estructuras de sintaxis usadas para predecir las muestras de bloques de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU.

[0044] El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 usa intrapredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU basándose en muestras descodificadas de la imagen asociada a la PU.

[0045] Si el codificador de vídeo 20 usa interpredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU basándose en muestras descodificadas de una o más imágenes distintas a la imagen asociada a la PU. El codificador de vídeo 20 puede usar unipredicción o bipredicción para generar los bloques predictivos de una PU. Cuando el codificador de vídeo 20 usa unipredicción para generar los bloques predictivos para una PU, la PU puede tener un único vector de movimiento (MV). Cuando el codificador de vídeo 20 usa bipredicción para generar los bloques predictivos para una PU, la PU puede tener dos MV.

[0046] Después de que el codificador de vídeo 20 genera bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para una o más PU de una CU, el codificador de vídeo 20 puede generar bloques residuales para la CU. Cada muestra en un bloque residual de la CU puede indicar una diferencia entre una muestra en un bloque predictivo de una PU de la CU y una muestra correspondiente en un bloque de codificación de la CU. Por ejemplo, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra en el bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques de predicción de luma de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual de Cb para la CU. Cada muestra en el bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb en uno de los bloques predictivos de Cb de la CU y una muestra correspondiente en el bloque de codificación de Cb original de la CU. El codificador de vídeo 20 también puede generar un bloque residual de Cr para la CU. Cada muestra en el bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques predictivos de Cr de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.

[0047] Además, el codificador de vídeo 20 puede usar la partición en árbol cuaternario para descomponer los bloques residuales (por ejemplo, los bloques residuales de luma, Cb y Cr) de una CU en uno o más bloques de transformada (por ejemplo, bloques de transformada de luma, Cb y Cr). Un bloque de transformada puede ser un bloque rectangular de muestras en el que se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede ser un bloque de transformada de muestras de luma, dos bloques de transformada correspondientes de muestras de croma y estructuras de sintaxis usadas para transformar las muestras de bloques de transformada. Por tanto, cada TU de una CU puede estar asociada a un bloque de transformada de luma, un bloque de transformada de Cb y un bloque de transformada de Cr. El bloque de transformada de luma asociado a la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformada de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un subbloque del bloque residual de Cr de la CU.

[0048] El codificador de vídeo 20 puede aplicar una o más transformaciones a un bloque de transformada para generar un bloque de coeficientes para una TU. Un bloque de coeficientes puede ser una matriz bidimensional de

coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. Por ejemplo, el codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU para generar un bloque de coeficientes de luma para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cb de una TU para generar un bloque de coeficientes de Cb para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

[0049] Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luma, un bloque de coeficientes de Cb o un bloque de coeficientes de Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación se refiere, en general, a un proceso en el que los coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos de sintaxis que indiquen los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar una codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos de sintaxis que indican los coeficientes de transformada cuantificados.

[0050] Con respecto a la CABAC, como ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden seleccionar un modelo de probabilidad (también denominado modelo de contexto) para codificar símbolos asociados a un bloque de datos de vídeo basándose en el contexto. Por ejemplo, un modelo de contexto (Ctx) puede ser un índice o desplazamiento que se aplica para seleccionar uno de una pluralidad de contextos diferentes, cada uno de los cuales puede corresponder a un modelo de probabilidad en particular. Por consiguiente, típicamente se define un modelo de probabilidad diferente para cada contexto. Después de codificar o descodificar el bin, el modelo de probabilidad se actualiza adicionalmente basándose en el valor del bin para reflejar las estimaciones de probabilidad más actuales para el bin. Por ejemplo, un modelo de probabilidad puede mantenerse como un estado en una máquina de estados finitos. Cada estado en particular puede corresponder a un valor de probabilidad específico. El siguiente estado, que corresponde a una actualización del modelo de probabilidad, puede depender del valor del bin actual (por ejemplo, el bin que se está codificando actualmente). Por consiguiente, la selección de un modelo de probabilidad puede estar influida por los valores de los bins previamente codificados, porque los valores indican, al menos en parte, la probabilidad de que el bin tenga un valor dado. El proceso de codificación del contexto, descrito anteriormente, puede denominarse en general un modo de codificación adaptativa al contexto.

[0051] Por lo tanto, el codificador de vídeo 20 puede codificar un símbolo de destino usando un modelo de probabilidad. Asimismo, el descodificador de vídeo 30 puede analizar un símbolo de destino usando el modelo de probabilidad. En algunos casos, el codificador de vídeo 20 puede codificar elementos de sintaxis usando una combinación de codificación adaptativa al contexto y no adaptativa al contexto. Por ejemplo, el codificador de vídeo 20 puede codificar bins por contexto seleccionando un modelo de probabilidad o "modelo de contexto" que opera sobre el contexto para codificar los bins. Por el contrario, el codificador de vídeo 20 puede codificar por omisión los bins evitando u omitiendo el proceso de codificación aritmética normal cuando se codifican los bins. En dichos ejemplos, el codificador de vídeo 20 puede usar un modelo de probabilidad fijo para codificar por omisión los bins. Es decir, los bins codificados por omisión no incluyen actualizaciones de contexto o probabilidad.

[0052] El codificador de vídeo 20 puede emitir un flujo de bits que incluye los elementos de sintaxis codificados por entropía. El flujo de bits también puede incluir elementos de sintaxis que no estén codificados por entropía. El flujo de bits puede incluir una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL incluye una cabecera de unidad NAL y encapsula una carga útil de secuencia de octetos sin procesar (RBSP). La cabecera de unidad NAL puede incluir un elemento de sintaxis que indica un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por la cabecera de unidad NAL de una unidad NAL indica el tipo de la unidad NAL. Una RBSP puede ser una estructura de sintaxis que contiene un número entero de octetos que están encapsulados dentro de una unidad NAL. En algunos casos, una RBSP incluye cero bits.

[0053] Diferentes tipos de unidades NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad de NAL puede encapsular una RBSP para un conjunto de parámetros de imagen (PPS), un segundo tipo de unidad de NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad de NAL puede encapsular una RBSP para información de mejora complementaria (SEI), y así sucesivamente. Las unidades NAL que encapsulan las RBSP para datos de codificación de vídeo (en contraposición a las RBSP para conjuntos de parámetros y mensajes SEI) pueden denominarse unidades NAL de capa de codificación de vídeo (VCL).

[0054] El descodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede analizar el flujo de bits para descodificar elementos de sintaxis del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose, al menos en parte, en los elementos de sintaxis descodificados a partir del flujo de bits. El proceso para reconstruir los datos de vídeo puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20. Por ejemplo, el descodificador de vídeo 30 puede usar los MV de las PU para determinar bloques predictivos para las PU de una

CU actual. Además, el descodificador de vídeo 30 puede cuantificar inversamente bloques de coeficientes de transformada asociados a las TU de la CU actual. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes de transformada para reconstruir bloques de transformada asociados a las TU de la CU actual. El descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a las muestras correspondientes de los bloques de transformada de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

[0055] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar configurados para realizar una codificación basada en paletas. Por ejemplo, en la codificación basada en paletas, en lugar de realizar las técnicas de codificación intrapredictiva o interpredictiva descritas anteriormente, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar una denominada paleta como una tabla de colores o de valores de píxeles que representan los datos de vídeo de un área en particular (por ejemplo, un bloque dado). De esta manera, en lugar de codificar los valores de píxeles reales o sus residuales para un bloque actual de datos de vídeo, el codificador de vídeo puede codificar valores de índices para uno o más de los valores de píxeles del bloque actual, donde los valores de índices indican entradas en la paleta que se usan para representar los valores de píxeles del bloque actual.

[0056] En el ejemplo anterior, el codificador de vídeo 20 puede codificar un bloque de datos de vídeo determinando una paleta para el bloque, ubicando una entrada de la paleta para representar el valor de cada píxel y codificando la paleta con valores de índices para los píxeles relacionando el valor del píxel con la paleta. El descodificador de vídeo 30 puede obtener, a partir de un flujo de bits codificado, una paleta para un bloque, así como valores de índices para los píxeles del bloque. El descodificador de vídeo 30 puede relacionar los valores de índices de los píxeles individuales con las entradas de paleta para reconstruir los valores de píxeles del bloque. En los casos en que el valor del índice asociado con un píxel individual no coincide con ningún valor del índice de la paleta correspondiente para el bloque, el descodificador de vídeo 30 puede identificar dicho píxel como un píxel de escape, para los propósitos de la codificación basada en paletas.

[0057] En otro ejemplo, el codificador de vídeo 20 puede codificar un bloque de datos de vídeo de acuerdo con las siguientes operaciones. El codificador de vídeo 20 puede determinar los valores residuales de predicción para píxeles individuales del bloque, determinar una paleta para el bloque y ubicar una entrada (por ejemplo, un valor del índice) en la paleta que tenga un valor representativo del valor de uno o más de los valores residuales de predicción de los píxeles individuales. Adicionalmente, el codificador de vídeo 20 puede codificar el bloque con valores de índices que indiquen la entrada en la paleta usada para representar el valor residual de predicción correspondiente para cada píxel individual del bloque. El descodificador de vídeo 30 puede obtener, a partir de un flujo de bits codificado, señalado por el dispositivo de origen 12, una paleta para un bloque, así como valores de índices para los valores residuales de predicción correspondientes a los píxeles individuales del bloque. Como se describe, los valores de índices pueden corresponder a entradas en la paleta asociada con el bloque actual. A su vez, el descodificador de vídeo 30 puede relacionar los valores de índices de los valores residuales de predicción con las entradas de la paleta para reconstruir los valores residuales de predicción del bloque. Los valores residuales de predicción pueden añadirse a los valores de predicción (por ejemplo, obtenidos usando intra o interpredicción) para reconstruir los valores de píxeles del bloque.

[0058] Como se describe con más detalle a continuación, la idea básica de la codificación basada en paletas es que, para un bloque dado de datos de vídeo que se vaya a codificar, el codificador de vídeo 20 puede obtener una paleta que incluya los valores de píxeles más dominantes en el bloque actual. Por ejemplo, la paleta puede referirse a un número de valores de píxeles que se determina o se supone que son dominantes y/o representativos de la CU actual. El codificador de vídeo 20 puede transmitir primero el tamaño y los elementos de la paleta al descodificador de vídeo 30. Adicionalmente, el codificador de vídeo 20 puede codificar los valores de píxeles en el bloque dado de acuerdo con un cierto orden de barrido. Para cada píxel incluido en el bloque dado, el codificador de vídeo 20 puede señalar el valor del índice que correlaciona el valor del píxel con una entrada correspondiente en la paleta. Si el valor del píxel no está incluido en la paleta (es decir, no existe una entrada de paleta que especifique un valor de píxel en particular del bloque codificado por la paleta), dicho píxel se define como un "píxel de escape". De acuerdo con la codificación basada en paletas, el codificador de vídeo 20 puede codificar y señalar un valor del índice que esté reservado para un píxel de escape. En algunos ejemplos, el codificador de vídeo 20 también puede codificar y señalar el valor del píxel o un valor residual (o versiones cuantificadas del mismo) para un píxel de escape incluido en el bloque dado.

[0059] Al recibir el flujo de bits de vídeo codificado señalado por el codificador de vídeo 20, el descodificador de vídeo 30 puede determinar primero la paleta basándose en la información recibida del codificador de vídeo 20. El descodificador de vídeo 30 puede asignar entonces los valores de índices recibidos asociados con las ubicaciones de píxeles en el bloque dado a las entradas de la paleta para reconstruir los valores de píxeles del bloque dado. En algunos casos, el descodificador de vídeo 30 puede determinar que un píxel de un bloque codificado por la paleta es un píxel de escape, como puede ser determinando que el píxel está codificado por la paleta con un valor del índice reservado para los píxeles de escape. En los casos en que el descodificador de vídeo 30 identifica un píxel de escape en un bloque codificado por la paleta, el descodificador de vídeo 30 puede recibir el valor del píxel o un valor

residual (o versiones cuantificadas del mismo) para un píxel de escape incluido en el bloque dado. El descodificador de vídeo 30 puede reconstruir el bloque codificado por la paleta correlacionando los valores de píxeles individuales con las entradas correspondientes de la paleta y utilizando el valor del píxel o el valor residual (o las versiones cuantificadas de los mismos) para reconstruir cualquier píxel de escape incluido en el bloque codificado por la paleta.

[0060] Como se ha indicado anteriormente, en un ejemplo de modo de codificación de la paleta, una paleta puede incluir entradas numeradas por un índice. Cada entrada puede representar valores o intensidades de componentes de color (por ejemplo, en los espacios de color como YCbCr, RGB, YUV, CMYK u otros formatos), que se pueden usar como un predictor para un bloque o como muestras finales de bloques reconstruidos. Como se describe en el documento de presentación estándar JCTVC-Q0094 (Wei Pu *et al.*, "AHG10: Software for Palette Coding based on RExt6.0," JCTVC-Q0094, Valencia, ES, del 27 de marzo al 4 de abril de 2014), una paleta puede incluir entradas que se copian desde una paleta predictora. Una paleta predictora puede incluir entradas de paleta de bloques codificados previamente usando el modo de paleta u otras muestras reconstruidas. Para cada entrada en la paleta predictora, se envía un indicador binario para indicar si esa entrada se copia en la paleta actual (indicada por el indicador = 1). Esto se denomina el vector de predicción de paleta binario. Además, la paleta actual puede comprender (por ejemplo, consistir en) nuevas entradas señalizadas explícitamente. El número de nuevas entradas también puede ser señalizado.

[0061] Para mencionar otro ejemplo, en el modo de paleta, una paleta puede incluir entradas numeradas por un índice que representa valores de componentes de color que pueden usarse como predictores para las muestras de bloques o como muestras finales de bloques reconstruidos. Cada entrada en la paleta puede contener, por ejemplo, un componente de luma (por ejemplo, valor de luma), dos componentes de croma (por ejemplo, dos valores de croma) o tres componentes de color (por ejemplo, RGB, YUV, etc.). Las entradas de paleta previamente descodificadas pueden almacenarse en una lista. Esta lista puede usarse para predecir entradas de paleta en el modo de paleta actual CU, por ejemplo. Se puede señalar un vector de predicción binario en el flujo de bits para indicar qué entradas de la lista se reutilizan en la paleta actual. En algunos ejemplos, la codificación de longitud de ejecución puede usarse para comprimir el predictor de paleta binaria. Por ejemplo, un valor de longitud de ejecución puede codificarse utilizando el código Exp-Golomb de orden 0.

[0062] En esta divulgación, se supone que cada entrada de paleta especifica los valores para todos los componentes de color de una muestra. Sin embargo, los conceptos de esta divulgación son aplicables al uso de una paleta independiente y/o una paleta independiente para cada componente de color. También, se supone que las muestras en un bloque se procesan usando un orden de barrido de trama horizontal. Sin embargo, también son aplicables otros barridos, como un orden de barrido de trama vertical. Como se ha mencionado anteriormente, una paleta puede contener entradas de paleta predichas, por ejemplo, predichas a partir de la(s) paleta(s) utilizada(s) para codificar el(los) bloque(s) anterior(es), y las nuevas entradas que pueden ser específicas para el bloque actual y se señalizan explícitamente. El codificador y el descodificador pueden conocer el número de entradas de paleta nuevas y predichas y una suma de ellas puede indicar el tamaño total de la paleta en un bloque.

[0063] Como se propone en el ejemplo de JCTVC-Q0094 citado anteriormente, cada muestra en un bloque codificado con la paleta puede pertenecer a uno de los tres modos, como se expone a continuación:

- Modo de escape: en este modo, el valor de muestra no se incluye en una paleta como una entrada de paleta y el valor de muestra cuantificado se señala explícitamente para todos los componentes de color. Es similar a la señalización de las nuevas entradas de paleta, aunque para las nuevas entradas de paleta, los valores de los componentes de color no están cuantificados.
- Modo de CopiaArriba (también llamado modo de CopiarDesdeArriba o modo de copia). En este modo, el índice de entrada de paleta para la muestra actual se copia de la muestra ubicada directamente arriba de la muestra actual en un bloque de muestras. En otros ejemplos, para el modo de copia arriba, se puede transponer un bloque de datos de vídeo de modo que la muestra sobre el bloque sea en realidad la muestra a la izquierda del bloque.
- Modo de valor (también llamado modo de índice o modo de ejecución). En este modo, el valor del índice de entrada de paleta se señala explícitamente.

[0064] Como se describe en el presente documento, un índice de entrada de paleta se puede mencionar como un índice de la paleta o simplemente un índice. Estos términos pueden usarse indistintamente para describir las técnicas de esta divulgación. Además, como se describe más en detalle a continuación, un índice de la paleta puede tener uno o más valores de color o intensidad asociados. Por ejemplo, un índice de la paleta puede tener un único valor asociado de color o intensidad, asociado a un único componente de color o intensidad de un píxel (por ejemplo, un componente rojo de datos RGB, un componente Y de datos YUV o similares). En otro ejemplo, un índice de la paleta puede tener múltiples valores de color o intensidad asociados. En algunos casos, la codificación basada en paletas se puede aplicar para codificar vídeo monocromo. Por consiguiente, el "valor de color" puede referirse en general a cualquier componente de color, o no de color, utilizado para generar un valor de píxel.

- 5 **[0065]** Un valor de ejecución puede indicar una ejecución de valores de índices de paleta que están codificados usando el mismo modo de codificación de la paleta. Por ejemplo, con respecto a el modo de valor, un codificador de vídeo (codificador de vídeo 20 o descodificador de vídeo 30) puede codificar un índice de la paleta y un valor de ejecución que indica un número de muestras consecutivas en un orden de barrido que tienen el mismo índice de la paleta y que están siendo codificadas con el índice de la paleta. Con respecto al modo CopiarDesdeArriba, el codificador de vídeo puede codificar una indicación de que un valor del índice para el valor de muestra actual se copia es el mismo que un valor del índice de una muestra vecina por arriba (por ejemplo, una muestra que está situada arriba de la muestra que está siendo codificada actualmente en un bloque) y un valor de ejecución que indica un número de muestras consecutivas en un orden de barrido que también copian un índice de la paleta de una muestra vecina por arriba y que está siendo codificada con el índice de la paleta. Por consiguiente, en los ejemplos anteriores, una ejecución de valores de índices de paleta se refiere a una ejecución de valores de paleta que tienen el mismo valor o una ejecución de valores de índices que se copian de las muestras vecinas por arriba.
- 10
- 15 **[0066]** Por lo tanto, la ejecución puede especificar, para un modo dado, el número de muestras posteriores que pertenecen al mismo modo. En algunos casos, la señalización de un índice y un valor de ejecución puede ser similar a la codificación de longitud de ejecución. En un ejemplo con fines ilustrativos, una cadena de valores de índices de paleta consecutivos de un bloque de índices correspondientes a un bloque de datos de vídeo puede ser 0, 2, 2, 2, 5. En algunos ejemplos, el bloque de índices puede incluir uno o más valores de píxeles de escape. Cada valor del índice en el bloque de índices puede corresponder a una muestra en el bloque de datos de vídeo. En este ejemplo, un codificador de vídeo puede codificar la segunda muestra (por ejemplo, el primer valor del índice de la paleta de "2") usando el modo de valor. Después de codificar un valor del índice de 2, el codificador de vídeo puede codificar una ejecución de 3, lo que indica que las tres muestras posteriores también tienen el mismo valor del índice de la paleta de 2. De manera similar, codificar una ejecución de cuatro índices de paleta después de codificar un índice usando el modo CopiarDesdeArriba puede indicar que se copian un total de cinco índices de paleta a partir de los valores de índices de paleta correspondientes en la fila que está arriba de la posición de muestra que se está codificando actualmente.
- 20
- 25
- 30 **[0067]** Usando la paleta, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden configurarse para codificar un bloque de muestras (por ejemplo, un bloque de datos de vídeo) en un bloque de índices, donde el bloque de índices es un bloque que incluye valores de índices, por ejemplo, para cada muestra, que asigna las muestras a una o más entradas de paleta y, en algunos ejemplos, incluye uno o más valores de píxeles de escape. Cada píxel de un bloque de datos de vídeo puede codificarse con el modo de ejecución, el modo de copia o el modo de escape. En algunos ejemplos, los píxeles en la primera fila del bloque de datos de vídeo solo pueden codificarse usando el modo de ejecución o el modo de escape.
- 35
- 40 **[0068]** El elemento de sintaxis *palette_run_type_flag* indica si se utiliza el modo de ejecución o el modo de copia. Por ejemplo, el codificador de vídeo 20 puede configurarse para señalar el elemento de sintaxis *palette_run_type_flag* codificando un valor correspondiente al elemento de sintaxis *palette_run_type_flag* en un flujo de bits codificado para una muestra de un bloque de datos de vídeo. El descodificador de vídeo 20 puede configurarse para recibir el flujo de bits codificado que comprende el valor codificado correspondiente al elemento de sintaxis *palette_run_type_flag*. El descodificador de vídeo 20 puede configurarse para descodificar el valor codificado para determinar el valor correspondiente al elemento de sintaxis *palette_run_type_flag* y, por lo tanto, determinar si se utiliza el modo de ejecución o el modo de copia para la muestra del bloque de datos de vídeo. Por ejemplo, cuando el valor de *palette_run_type_flag* es un primer valor, se puede usar el modo de ejecución para la muestra del bloque de datos de vídeo. Para mencionar otro ejemplo, cuando el valor de *palette_run_type_flag* es un segundo valor, se puede usar el modo de copia para la muestra del bloque de datos de vídeo.
- 45
- 50 **[0069]** En algunos ejemplos, cuando se utiliza el modo de ejecución o el modo de copia, el elemento de sintaxis *palette_index* puede señalizarse junto con el elemento de sintaxis *palette_run*. Por ejemplo, el codificador de vídeo 20 puede configurarse para señalar los elementos de sintaxis *palette_index* y *palette_run* codificando un valor (por ejemplo, un valor del índice) correspondiente a *palette_index* y un valor (por ejemplo, un valor de ejecución) correspondiente a *palette_run* en un flujo de bits codificado. El descodificador de vídeo 30 puede configurarse para recibir el flujo de bits codificado que comprende el valor codificado correspondiente al elemento de sintaxis *palette_index* y el valor codificado correspondiente al elemento de sintaxis *palette_run*. El descodificador de vídeo 20 puede configurarse para descodificar el valor codificado correspondiente a *palette_index* y el valor codificado correspondiente a *palette_run* para determinar respectivamente el valor (por ejemplo, valor del índice) correspondiente a *palette_index* y el valor (por ejemplo, valor de ejecución) correspondiente a *palette_run*.
- 55
- 60 **[0070]** Cuando se utiliza el modo de ejecución, el valor de ejecución indica el número de píxeles que tendrán el mismo índice de la paleta. Sin embargo, cuando se utiliza el modo de copia, el valor de ejecución indica el número de píxeles para los cuales se copia el índice de la paleta (por ejemplo, el valor del índice) de otro píxel respectivo a cada píxel (por ejemplo, directamente arriba de cada píxel respectivo).
- 65 **[0071]** En algunos ejemplos, el modo de escape se codifica dentro del modo de ejecución, donde se puede usar un índice de la paleta específico para indicar este modo. El índice de la paleta utilizado para indicar el modo de

escape es igual al tamaño de la paleta del bloque actual según algunos ejemplos. En el modo de escape, el valor de ejecución puede no estar codificado ya que el modo de escape se aplica a un solo píxel (por ejemplo, un triplete de píxeles (Y, U y V)) donde los valores de los componentes de color para los píxeles individuales se señalizan explícitamente como *palette_escape_val*. En algunos ejemplos, el modo de copia puede no estar habilitado para la primera fila del bloque dado que no hay píxeles arriba de la primera fila que pertenezcan al mismo bloque.

[0072] Un identificador *palette_escape_val_present_flag* puede señalizarse por bloque para indicar el uso de los píxeles de escape. Este indicador es igual a 1 indica que hay al menos un píxel de escape en el bloque codificado por la paleta y de lo contrario el indicador es igual a 0. Por ejemplo, el codificador de vídeo 20 puede configurarse para señalar el elemento de sintaxis *palette_escape_val_present_flag* codificando un valor correspondiente al elemento de sintaxis *palette_escape_val_present_flag* en un flujo de bits codificado. El descodificador de vídeo 20 puede configurarse para recibir el flujo de bits codificado que comprende el valor codificado correspondiente al elemento de sintaxis *palette_escape_val_present_flag*. El descodificador de vídeo 20 puede configurarse para descodificar el valor codificado para determinar el valor correspondiente al elemento de sintaxis *palette_escape_val_present_flag* y, por lo tanto, determinar si hay al menos un píxel de escape en el bloque codificado por la paleta.

[0073] En algunos ejemplos, el tamaño de la paleta está restringido para estar en el intervalo de 0 a *max_palette_size* con el último que está señalizado. Para un bloque codificado con modo de paleta, la paleta puede, en algunos ejemplos, predecirse a partir de las entradas de la paleta de uno o más bloques previamente codificados por la paleta. La paleta se puede ser señalizada explícitamente para un bloque actual como una o más entradas nuevas. En otros ejemplos, la paleta de un bloque previamente codificado puede reutilizarse por completo (por ejemplo, copiarse) para el bloque actual, lo que se denomina modo de compartición de paleta. En algunos ejemplos, se puede señalar un identificador *palette_share_flag* para indicar que la paleta entera del bloque anterior se reutiliza sin modificación tal como está para el bloque actual.

[0074] Al codificar un bloque de vídeo usando el modo de paleta, el patrón de barrido de píxeles (por ejemplo, orden de barrido) puede incluir, por ejemplo: barrido con recorrido vertical o recorrido horizontal (como una serpiente). El patrón de barrido utilizado en el bloque puede derivarse de acuerdo con el identificador *palette_transpose_flag* señalizado por unidad de bloque.

[0075] Durante la codificación del modo de paleta, se puede aplicar un proceso de ajuste del índice de la paleta. A partir del segundo píxel en el bloque actual, se puede verificar el modo de paleta del píxel anterior en el orden de barrido (por ejemplo, determinado). En algunos ejemplos, el tamaño máximo del índice de la paleta se puede reducir primero en 1. Si el modo de paleta para el píxel anterior en el orden de barrido es igual al modo de ejecución (es decir, si el píxel anterior en el orden de barrido ha sido o será codificado usando el modo de ejecución), el índice de la paleta (por ejemplo, el valor del índice) para el píxel actual puede reducirse en 1 si el valor del índice es mayor o igual que el valor del índice para el píxel anterior en el orden de barrido. De manera similar, si el modo de paleta para el píxel anterior en el orden de barrido es igual al modo de copia (es decir, si el píxel anterior en el orden de barrido ha sido o será codificado usando el modo de copia), entonces el índice de la paleta (por ejemplo, el valor del índice) para el píxel actual puede reducirse en 1 si el índice es mayor que el índice de la paleta de arriba.

[0076] El codificador de vídeo 20 puede configurarse para codificar por entropía el bloque de índices para comprimir el bloque de índices. De manera similar, el descodificador de vídeo 30 puede configurarse para descodificar por entropía un bloque de índices codificado para generar el bloque de índices desde el cual el descodificador de vídeo 30 puede generar un bloque de muestras (por ejemplo, el bloque de datos de vídeo codificado por el codificador 20). Por ejemplo, la codificación por entropía basada en la longitud de ejecución puede usarse para comprimir y descomprimir el bloque de índices. En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para codificar y descodificar respectivamente los valores de índices en el bloque de índices usando CABAC.

[0077] Para aplicar la codificación CABAC a la información (por ejemplo, un elemento de sintaxis, un bloque de índices tal como los valores de índices del bloque de índices u otra información), un codificador de vídeo (por ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30) puede realizar la binarización de la información. La binarización se refiere al proceso de convertir información en una serie de uno o más bits. Cada serie de uno o más bits puede denominarse "bins". La binarización es un proceso sin pérdidas y puede incluir una sola o una combinación de las siguientes técnicas de codificación: codificación de longitud fija, codificación unaria, codificación unaria truncada, codificación de Rice truncada, codificación de Golomb, codificación de Golomb exponencial, codificación de Golomb-Rice, cualquier codificación de Golomb, cualquier codificación de Rice, y cualquier forma de codificación por entropía. Por ejemplo, la binarización puede incluir representar el valor del número entero de 5 como 00000101 usando una técnica de longitud fija de 8 bits o como 11110 usando una técnica de codificación unaria.

[0078] Después de la binarización, un codificador de vídeo puede identificar un contexto de codificación. El contexto de codificación puede identificar probabilidades de codificación de bins que tengan valores particulares. Por ejemplo, un contexto de codificación puede indicar una probabilidad de 0,7 de codificar un bin de valor 0 y una

probabilidad de 0,3 de codificar un bin de valor 1. Después de identificar el contexto de codificación, el codificador de vídeo puede codificar aritméticamente ese bin basándose en el contexto, lo cual se conoce como codificación en modo de contexto. Los bins codificados usando una codificación en modo de contexto CABAC pueden denominarse "bins de contexto".

5
 [0079] Además, en lugar de realizar la codificación en modo de contexto en todos los bins, un codificador de vídeo (por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30) puede codificar algunos bins usando la codificación CABAC de derivación (por ejemplo, la codificación en modo de derivación). La codificación en modo de derivación se refiere a un modo de derivación de un codificador CABAC, donde la codificación de derivación es el proceso de codificar aritméticamente un bin sin utilizar un contexto adaptativo (por ejemplo, un contexto de codificación). Es decir, el motor de codificación de derivación no selecciona contextos y puede suponer una probabilidad de 0,5 para ambos símbolos (0 y 1). Aunque la codificación en modo de derivación puede no ser tan eficaz en lo que respecta al ancho de banda como la codificación en modo de contexto, llevar a cabo la codificación en modo de derivación en un bin puede ser computacionalmente menos caro que llevar a cabo una codificación en modo de contexto en el bin. Además, llevar a cabo la codificación en modo de derivación puede permitir un mayor grado de paralelización y rendimiento. Los bins codificados usando la codificación en modo de derivación pueden denominarse "bins de derivación".

20
 [0080] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse con un codificador CABAC (por ejemplo, un codificador CABAC y un decodificador CABAC, respectivamente). Un codificador CABAC puede incluir un motor de codificación en modo de contexto para realizar la codificación en modo de contexto y un motor de codificación en modo de derivación para realizar la codificación en modo de derivación. Si un bin se codifica en modo de contexto, se usa el motor de codificación en modo de contexto para codificar este bin. El motor de codificación en modo de contexto puede necesitar más de dos ciclos de procesamiento para codificar un único bin. Sin embargo, con un diseño de canalización adecuado, un motor de codificación en modo de contexto puede necesitar solamente $n+M$ ciclos para codificar n bins, donde M es la sobrecarga para iniciar la canalización. Normalmente, M es mayor que 0.

30
 [0081] Al inicio del proceso de codificación CABAC (es decir, en cada conmutación del modo de derivación al modo de contexto y viceversa), se introduce la sobrecarga de la canalización. Si un bin se codifica en modo de derivación, se usa el motor de codificación en modo de derivación para codificar este bin. Cabe esperar que el motor de codificación en modo de derivación necesite solamente un ciclo para codificar n bits de información, donde n puede ser mayor que uno. Por tanto, el número total de ciclos para codificar un conjunto de bins de derivación y bins de contexto puede reducirse si todos los bins de derivación dentro del conjunto se codifican juntos (por ejemplo, en secuencia sin bins codificados en contexto intercalados), y todos los bins de contexto dentro del conjunto se codifican juntos (por ejemplo, en secuencia sin bins codificados en derivación intercalados). En particular, codificar juntos los bins de derivación antes o después de efectuar la transición a la codificación en modo de contexto puede ahorrar la sobrecarga requerida para volver a iniciar el motor de codificación en modo de contexto. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para conmutar del modo de derivación al modo de contexto (o del modo de contexto al modo de derivación en otros ejemplos) una sola vez, a través de una serie de bins codificados en derivación y en contexto, al mismo tiempo que codifican y decodifican respectivamente un bloque de datos de vídeo usando el modo de paleta. En otro ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para reducir el número de veces que el proceso de codificación o decodificación conmuta del modo de derivación al modo de contexto (y del modo de contexto al modo de derivación) al codificar o decodificar un bloque de datos de vídeo usando el modo de paleta.

50
 [0082] Las técnicas descritas en esta divulgación también pueden incluir técnicas para diversas combinaciones de uno o más entre señalización de modos de codificación de vídeo basados en paletas, transmisión de paletas, derivación de paletas, señalización del orden de barrido, señalización del orden de barrido y transmisión de mapas de codificación de vídeo basados en paletas y otros elementos de sintaxis. Por ejemplo, las técnicas de esta divulgación pueden dirigirse a la información de la paleta de codificación por entropía. En algunos ejemplos, las técnicas de esta divulgación pueden, entre otras cosas, usarse para aumentar la eficacia de la codificación y reducir las ineficacias de la codificación asociadas a la codificación de vídeo basada en paletas. Por consiguiente, como se describe más en detalle a continuación, las técnicas de esta divulgación pueden, en algunos casos, mejorar la eficacia y mejorar la tasa de bits cuando se codifican datos de vídeo usando un modo de paleta.

60
 [0083] Las técnicas, aspectos y/o ejemplos descritos en el presente documento pueden utilizarse en conjunción entre sí en cualquier combinación o por separado uno del otro. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para realizar cualquiera o cualquier combinación adecuada de una o más de las técnicas, aspectos y/o ejemplos descritos en el presente documento.

65
 [0084] Un problema con el sistema de codificación de ejemplo se describe en el documento de Tzu-Der Chuang *et al.*, "CE-1 related: Index Map scan for 64 x 64 palette coding block", JCTVC-T0058 versión 3, cargado en el sistema de gestión de documentos JCT-VC el 10 de febrero de 2015 (en lo sucesivo "JCTVC-T0058") es que el tamaño de bloque de la paleta puede ser tan grande como 64 x 64 y con un patrón de barrido tan grande como 64 x 64, pero el tamaño más grande de bloque de transformada es 32 x 32 donde, por ejemplo, se aplica el barrido de coeficientes.

Entonces, en este caso, la canalización en la implementación se incrementará a 64 x 64 tamaños de bloque, lo cual no es necesario sin el modo de paleta y, por lo tanto, presenta un caso especial para el modo de paleta. En JCTVC-T0058 se describe la codificación de un bloque de 64 x 64 en modo de paleta como cuatro subbloques de 32 x 32 alterando el recorrido de un barrido de 64 x 64 en el recorrido de cuatro barridos de 32 x 32. Sin embargo, hacerlo requeriría un cambio en la codificación del modo de paleta, que sería específico solo para un bloque de paleta de 64 x 64 y, por lo tanto, introduciría, por ejemplo, la no uniformidad en la codificación del modo de paleta.

[0085] Una versión posterior de JCTVC-T0058, concretamente, la versión 4, también divulga la opción de deshabilitar la codificación del modo de paleta para 64 x 64 bloques (unidades de codificación) todos a la vez.

[0086] En diversos ejemplos de esta divulgación, las técnicas de esta divulgación pueden dirigirse a procesos de predicción o codificación de un bloque en modo de paleta para mejorar la eficacia de la codificación y/o reducir la complejidad del códec, por ejemplo, abordando cómo, si es que lo hace, un bloque de 64 x 64 se codificará utilizando el modo de paleta.

[0087] En algunos ejemplos de esta divulgación, la codificación del modo de paleta puede deshabilitarse para cualquier bloque de paleta que tenga un tamaño de 64 x 64 o mayor. En otros ejemplos, la codificación del modo de paleta se puede restringir a bloques de paleta que tengan un tamaño inferior a 64 x 64, lo que significa que la codificación del modo de paleta se puede habilitar o, en cualquier caso, utilizar para bloques de paleta que tengan un tamaño inferior a 64 x 64. En otros ejemplos, el tamaño de bloque de la paleta más grande puede estar restringido normativamente en base al tamaño de la unidad de transformada más grande, como puede ser, estar restringido normativamente al tamaño de la unidad de transformada más grande. La codificación del modo de paleta puede deshabilitarse para un tamaño de bloque de la paleta que sobrepasa o, en cualquier caso, es mayor que el tamaño de la unidad de transformada más grande. En dichos ejemplos, se entiende que el tamaño de bloque de la paleta más grande puede basarse en el tamaño de la unidad de transformada más grande en que el tamaño de bloque de la paleta más grande está restringido normativamente al tamaño de la unidad de transformada más grande. Por ejemplo, el codificador de vídeo 20 puede configurarse para restringir normativamente el tamaño de bloque de la paleta más grande que puede codificarse usando el modo de paleta al tamaño de la unidad de transformada más grande. En este ejemplo, el codificador de vídeo 20 puede configurarse para deshabilitar el modo de paleta o, en cualquier caso, no usar el modo de paleta para ningún bloque de paleta que tenga un tamaño mayor que el tamaño de la unidad de transformada más grande que el codificador de vídeo 20 está configurado para codificar.

[0088] Por ejemplo, si el tamaño de la unidad de transformada más grande que el codificador de vídeo 20 está configurado para codificar es 32 x 32, entonces el codificador de vídeo 20 puede configurarse para restringir normativamente el tamaño de bloque de la paleta más grande a 32 x 32. En dicho ejemplo, el codificador de vídeo 20 puede configurarse para deshabilitar el modo de paleta o, en cualquier caso, no usar el modo de paleta para ningún bloque de paleta que tenga un tamaño mayor que 32 x 32. También se entiende en dicho ejemplo que el codificador de vídeo 20 puede configurarse para habilitar el modo de paleta o, en cualquier caso, usar el modo de paleta para cualquier bloque de paleta que tenga un tamaño menor o igual a 32 x 32. Los ejemplos de bloques de paleta que tienen un tamaño mayor que 32 x 32 incluyen, por ejemplo, 64 x 64, 64 x 16, 16 x 64, 64 x 32 y 32 x 64.

[0089] Para mencionar otro ejemplo, si el tamaño de la unidad de transformada más grande que el codificador de vídeo 20 está configurado para codificar es 16 x 16, entonces el codificador de vídeo 20 puede configurarse para restringir normativamente el tamaño de bloque de la paleta más grande a 16 x 16. En dicho ejemplo, el codificador de vídeo 20 puede configurarse para deshabilitar el modo de paleta o, en cualquier caso, no usar el modo de paleta para ningún bloque de paleta que tenga un tamaño mayor que 16 x 16. También se entiende en dicho ejemplo que el codificador de vídeo 20 puede configurarse para habilitar el modo de paleta o, en cualquier caso, usar el modo de paleta para cualquier bloque de paleta que tenga un tamaño menor o igual a 16 x 16.

[0090] En otros ejemplos, el tamaño de la unidad de transformada más grande que el codificador de vídeo 20 está configurado para codificar puede estar restringido normativamente a un tamaño de bloque de M x N, donde M y N son números enteros positivos y pueden o no ser iguales entre sí. En algunos ejemplos, M y/o N pueden basarse en el tamaño de la unidad de transformada más grande. Por ejemplo, si el tamaño de la unidad de transformada más grande es 32 x 32, entonces M y N equivaldrían a 32. Sin embargo, en un ejemplo donde el tamaño de la unidad de transformada más grande es 32 x 16, entonces M sería igual a 32 y N sería igual a 16. En dicho ejemplo, los ejemplos de bloques de paleta que tienen un tamaño mayor que 32 x 16 incluyen, por ejemplo, 64 x 64, 64 x 16, 16 x 64, 64 x 32, 32 x 64, 32 x 32 y 16 x 32.

[0091] En algunos ejemplos, el codificador de vídeo 20 puede configurarse para señalar el tamaño de la unidad de transformada más grande para un conjunto de datos en particular. En dichos ejemplos, el codificador de vídeo 20 puede configurarse para deshabilitar el modo de paleta o, en cualquier caso, no usar el modo de paleta para ningún bloque de paleta asociado al conjunto de datos en particular que tenga un tamaño de bloque mayor que el tamaño de la unidad de transformada más grande señalado. Por consiguiente, como se usa en el presente documento, la unidad de transformada más grande puede referirse a la unidad de transformada más grande que el codificador de vídeo 20 está configurado para codificar, o puede referirse a una unidad de transformada más grande señalizada

para un conjunto de datos en particular (por ejemplo, uno o más bloques de datos de vídeo). Por ejemplo, mientras que el tamaño de la unidad de transformada más grande puede ser 32 x 32, el codificador de vídeo 20 puede señalar, para un conjunto de datos en particular, que el tamaño de la unidad de transformada más grande es 16 x 16. Por lo tanto, para este conjunto de datos en particular en este ejemplo, el tamaño de la unidad de transformada más grande es 16 x 16.

[0092] Por consiguiente, se entiende que el codificador de vídeo 20 puede configurarse para deshabilitar dinámicamente el modo de paleta o, en cualquier caso, configurarse para no usar el modo de paleta basado en el tamaño de la unidad de transformada más grande. De manera similar, se entiende que el codificador de vídeo 20 puede configurarse para deshabilitar dinámicamente el modo de paleta o, en cualquier caso, configurarse para no usar el modo de paleta para ningún bloque de paleta que tenga un tamaño mayor que la unidad de transformada más grande. Por lo tanto, también se entiende que el codificador de vídeo 20 puede configurarse para deshabilitar dinámicamente el modo de paleta o, en cualquier caso, configurarse para no usar el modo de paleta para ningún bloque de paleta que tenga un tamaño que no sea igual o menor que la unidad de transformada más grande. Por lo tanto, se entiende además que el codificador de vídeo 20 puede configurarse para codificar un bloque de datos de vídeo usando el modo de paleta solo cuando el bloque de datos de vídeo tiene un tamaño que no sobrepasa la unidad de transformada más grande que el codificador de vídeo 20 puede configurarse para codificar. De manera similar, el codificador de vídeo 20 puede configurarse para habilitar la codificación del modo de paleta para un bloque de datos de vídeo solo cuando el bloque de datos de vídeo tiene un tamaño que no sobrepasa la unidad de transformada más grande.

[0093] Asimismo, se entiende que el descodificador de vídeo 30 puede configurarse para deshabilitar dinámicamente el modo de paleta o, en cualquier caso, configurarse para no usar el modo de paleta para ningún bloque de paleta que tenga un tamaño que no sea igual o menor que la unidad de transformada más grande. Por lo tanto, se entiende además que el descodificador de vídeo 30 puede configurarse para descodificar un bloque de datos de vídeo usando el modo de paleta solo cuando el bloque de datos de vídeo tiene un tamaño que no sobrepasa la unidad de transformada más grande que el codificador de vídeo 20 puede configurarse para codificar y/o ese descodificador de vídeo 30 puede configurarse para descodificar. De manera similar, el descodificador de vídeo 30 puede configurarse para habilitar la codificación del modo de paleta para un bloque de datos de vídeo solo cuando el bloque de datos de vídeo tiene un tamaño que no sobrepasa la unidad de transformada más grande. En otros ejemplos, el descodificador de vídeo 30 puede configurarse para determinar si el modo de paleta está habilitado o deshabilitado en base a un valor correspondiente al indicador de modo de paleta, tal como un valor para el elemento de sintaxis de `palette_mode_flag`.

[0094] Para mencionar otro ejemplo, el descodificador de vídeo 30 puede configurarse para recibir un bloque de datos de vídeo. El descodificador de vídeo 30 puede configurarse para determinar el tamaño de los datos de vídeo en bloque en relación con el tamaño de la unidad de transformada más grande. El descodificador de vídeo 30 puede configurarse para determinar que el bloque de vídeo recibido no está codificado en modo de paleta cuando el bloque de datos de vídeo recibido es mayor que el tamaño de la unidad de transformada más grande.

[0095] Como se expone en el presente documento, el tamaño de bloque de la paleta más grande puede estar restringido normativamente. Por ejemplo, el tamaño de bloque de la paleta más grande puede basarse en el tamaño de la unidad de transformada más grande, como puede ser, estar restringido normativamente al tamaño de la unidad de transformada más grande. En algunos ejemplos, el codificador de vídeo 20 puede configurarse con una restricción de flujo de bits de conformidad para implementar cualquier restricción en el tamaño de bloque de la paleta descrita en el presente documento que da como resultado el control cuando el modo de paleta está deshabilitado, habilitado o, en cualquier caso, utilizado. Por ejemplo, la restricción de conformidad con el flujo de bits puede ser que un flujo de bits de conformidad no tenga un bloque que sobrepase un cierto tamaño codificado con el modo de paleta. Para mencionar otro ejemplo, la restricción de conformidad con el flujo de bits puede ser que un flujo de bits de conformidad tenga un bloque codificado con el modo de paleta solo en el caso en que el bloque sea igual o menor que cierto tamaño. En ambos ejemplos, el cierto tamaño referenciado puede ser 32 x 32 o cualquier otro tamaño M x N, donde M y N son números enteros positivos y pueden o no ser iguales entre sí. Sin embargo, en otros ejemplos, el cierto tamaño referenciado en ambos ejemplos anteriores puede basarse en el tamaño de la unidad de transformada más grande. En dichos ejemplos, la restricción de conformidad con el flujo de bits puede, por ejemplo, ser que un flujo de bits de conformidad no tenga un bloque que sobrepase la unidad de transformada más grande. Para mencionar otro ejemplo, la restricción de conformidad con el flujo de bits puede ser que un flujo de bits de conformidad debe cumplir con una o más restricciones normativas descritas en el presente documento.

[0096] Con respecto a cualquier restricción de conformidad con el flujo de bits descrita en el presente documento, se entiende que el codificador de vídeo 20 puede configurarse con cualquiera de dichas restricciones en cualquier combinación para controlar cuando el modo de paleta está deshabilitado, habilitado o, en cualquier caso, utilizado para un bloque de datos de vídeo.

[0097] En otros ejemplos, el codificador de vídeo 20 puede configurarse para implementar cualquier restricción del tamaño de bloque de la paleta descrita en el presente documento al configurarse para dividir cualquier bloque de datos de vídeo para que el modo de paleta se codifique en subbloques de M x N de manera que todo el bloque de

datos de vídeo sea representado por subbloques de $M \times N$, donde M y N son números enteros positivos y pueden o no ser iguales entre sí. Dividir todo el bloque de datos de vídeo significa que cada píxel (por ejemplo, muestra) del bloque de datos de vídeo forma parte de un subbloque de $M \times N$. El tamaño del subbloque puede depender de uno o más criterios. Por ejemplo, el tamaño del subbloque de $M \times N$ puede depender del tamaño del bloque utilizado en la codificación de los coeficientes de transformada (por ejemplo, el tamaño de un bloque de transformada en una TU) para alinear la codificación del modo de paleta con la codificación de los coeficientes de transformada. En dicho ejemplo, si el codificador de vídeo 20 está configurado para transformar el código de los coeficientes usando bloques de 4×4 de tamaño, entonces el codificador de vídeo 20 puede configurarse para dividir cualquier bloque de datos de vídeo para que el modo de paleta se codifique en subbloques de 4×4 , donde M y N son iguales a 4. Por ejemplo, en lugar de codificar un bloque de 64×64 usando el modo de paleta, el codificador de vídeo 20 puede configurarse para dividir el bloque de 64×64 en una pluralidad de subbloques de 4×4 que dan como resultado doscientos cincuenta y seis subbloques de 4×4 en este ejemplo, con cada subbloque codificado individualmente usando el modo de paleta.

[0098] En otro ejemplo, en lugar de depender de uno o más criterios, el tamaño del subbloque de $M \times N$ puede ser un tamaño por defecto menor que 64×64 . Por ejemplo, el tamaño por defecto del subbloque de $M \times N$ puede ser 4×4 , 8×8 , 16×16 , 32×32 o cualquier otro tamaño inferior a 64×64 . En este ejemplo, el codificador de vídeo 20 puede configurarse para implementar cualquier restricción del tamaño de bloque de la paleta descrita en el presente documento al configurarse para dividir cualquier bloque de datos de vídeo para que el modo de paleta se codifique en el tamaño por defecto, como 4×4 , 8×8 , 16×16 , 32×32 , o cualquier otro tamaño menor que 64×64 , respectivamente.

[0099] En algunos ejemplos, los subbloques de $M \times N$ pueden barrerse de acuerdo con cualquier orden de barrido. Por ejemplo, el codificador de vídeo 20 puede configurarse para barrer los subbloques de $M \times N$ usando un orden de barrido en zigzag, un orden de barrido horizontal, un orden de barrido vertical, un orden de barrido "tipo serpiente" (es decir, un orden de barrido transversal), o cualquier otro orden de barrido.

[0100] En otros ejemplos, el codificador de vídeo 20 puede configurarse para implementar cualquier restricción del tamaño de bloque de la paleta descrita en el presente documento al configurarse para señalar el modo de paleta (por ejemplo, al señalar un valor para el elemento de sintaxis de `palette_mode_flag`) para bloques que tienen un tamaño de 64×64 , pero también está configurado para señalar otra información relacionada con la paleta (por ejemplo, entradas de paleta reutilizadas, entradas de paleta nuevas, tamaño de tabla de paleta, etc.) para tamaños de subbloque de $M \times N$ menores que 64×64 , donde M y N son números enteros positivos y pueden o no ser iguales entre sí. Por ejemplo, el tamaño del subbloque de $M \times N$ puede ser 32×32 . En algunos ejemplos, el tamaño del subbloque de $M \times N$ puede ser 32×32 porque 32×32 corresponde al tamaño de la unidad de transformada más grande. En dicho ejemplo, el codificador de vídeo 20 puede configurarse para implementar cualquier restricción del tamaño de bloque de la paleta descrita en el presente documento al configurarse para señalar el modo de paleta en bloques que tienen un tamaño de 64×64 , pero también configurarse para señalar otra información relacionada con la paleta en tamaños de subbloques de 32×32 (o cualquier otro tamaño de subbloque de $M \times N$). Este es un ejemplo descrito en el presente documento donde el codificador de vídeo 20 puede configurarse para armonizar el tamaño de bloque del modo de paleta con el tamaño de bloque de la unidad de transformada. En algunos ejemplos, M y/o N pueden basarse en el tamaño de la unidad de transformada más grande. Por ejemplo, si el tamaño de la unidad de transformada más grande es 32×32 , entonces M y N equivaldrían a 32. El orden de barrido para el tamaño de bloque de la paleta de 64×64 puede ser el mismo que el orden de barrido para cada uno de los bloques de $M \times N$.

[0101] En un ejemplo que implica un tamaño de bloque de la paleta de 64×64 , el codificador de vídeo puede configurarse para señalar el modo de paleta para este bloque de paleta de tamaño 64×64 . El codificador de vídeo puede configurarse para señalar otra información relacionada con la paleta para cada subbloque de $M \times N$. Por ejemplo, el codificador de vídeo 20 puede configurarse para señalar el `max_palette_size` para cada subbloque de $M \times N$.

[0102] En otros ejemplos, el codificador de vídeo 20 puede configurarse para implementar cualquier restricción del tamaño de bloque de la paleta descrita en el presente documento al configurarse para restringir la longitud de ejecución más larga de los valores de índices y/o valores de escape para que sea menor que un valor umbral de T . En dichos ejemplos, en lugar de dividir un bloque de paleta de 64×64 en subbloques, el codificador de vídeo 20 puede configurarse para limitar la longitud de ejecución máxima para que sea menor que el valor umbral de T . Al restringir el valor de la longitud de ejecución máxima, el codificador de vídeo 20 puede configurarse para implementar la restricción del tamaño de bloque de la paleta sin dividir el bloque de paleta en subbloques.

[0103] En algunos ejemplos, T puede ser igual al tamaño de la unidad de transformada más grande. Por ejemplo, si el tamaño de la unidad de transformada más grande es 32×32 , entonces T puede ser igual a 32×32 . Ahora se describe un ejemplo que implica un orden de barrido con recorrido horizontal para un bloque de paleta de 64×64 y un valor T de 32×32 . En lugar de procesar el bloque de paleta en 32×32 cuadrantes (por ejemplo, un valor del índice seguido de una longitud de ejecución menor que T , tal como 32×32 menos 1) en este ejemplo, el codificador

de vídeo 20 puede procesar el bloque de paleta de 64 x 64 como tal como está, pero limitar la longitud de ejecución máxima a un valor menor que el valor de T, tal como 32 x 32 menos 1.

5 **[0104]** En algunos ejemplos, el codificador de vídeo 20 puede configurarse con una restricción de flujo de bits de conformidad para implementar cualquier restricción del tamaño de bloque de la paleta descrita en el presente documento. Por ejemplo, la restricción de conformidad con el flujo de bits puede ser que un flujo de bits de conformidad no incluya un valor de longitud de ejecución igual o mayor que el valor umbral de T.

10 **[0105]** La FIG. 2 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que puede implementar las técnicas de esta divulgación. La FIG. 2 se proporciona con propósitos explicativos y no se debería considerar limitante de las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con propósitos explicativos, esta divulgación describe un codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

15 **[0106]** El codificador de vídeo 20 representa un ejemplo de dispositivo que puede estar configurado para realizar técnicas de codificación de vídeo basada en paletas de acuerdo con diversos ejemplos descritos en esta divulgación.

20 **[0107]** En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de codificación de bloque 100, una memoria de datos de vídeo 101, una unidad de generación residual 102, una unidad de procesamiento de transformada 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de procesamiento de transformada inversa 110, una unidad de reconstrucción 112, una unidad de filtro 114, una memoria intermedia de imágenes descodificadas 116 y una unidad de codificación por entropía 118. La unidad de codificación de bloque 100 incluye una unidad de procesamiento de interpretación 120 y una unidad de procesamiento de intrapredicción 126. La unidad de procesamiento de interpretación 120 incluye una unidad de estimación de movimiento y una unidad de compensación de movimiento (no se muestran). El codificador de vídeo 20 también incluye una unidad de codificación basada en paletas 122 configurada para realizar diversos aspectos de las técnicas de codificación basada en paletas descritas en esta divulgación. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales.

30 **[0108]** La memoria de datos de vídeo 101 puede almacenar datos de vídeo a codificar mediante los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 101 se pueden obtener, por ejemplo, a partir de la fuente de vídeo 18. La memoria intermedia de imágenes descodificadas 116 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo mediante el codificador de vídeo 20, por ejemplo, en los modos de intracodificación o intercodificación. La memoria de datos de vídeo 101 y la memoria intermedia de imágenes descodificadas 116 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El mismo dispositivo de memoria u otros dispositivos de memoria separados pueden proporcionar una memoria de datos de vídeo 101 y una memoria intermedia de imágenes descodificadas 116. En diversos ejemplos, la memoria de datos de vídeo 101 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera del chip con respecto a esos componentes.

45 **[0109]** El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede codificar cada CTU en un fragmento de una imagen de los datos de vídeo. Cada una de las CTU puede estar asociada a bloques de árbol de codificación (CTB) de luma de igual tamaño y a CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de codificación de bloque 100 puede realizar una partición de árbol cuaternario para dividir los CTB de la CTU en bloques progresivamente más pequeños. Los bloques más pequeños pueden ser bloques de codificación de las CU. Por ejemplo, la unidad de codificación de bloque 100 puede dividir un CTB asociado a una CTU en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subbloques de igual tamaño, y así sucesivamente.

50 **[0110]** El codificador de vídeo 20 puede codificar las CU de una CTU para generar representaciones codificadas de las CU (es decir, CU codificadas). Como parte de la codificación de una CU, la unidad de codificación de bloque 100 puede dividir los bloques de codificación asociados a la CU entre una o más PU de la CU. Por tanto, cada PU puede estar asociada a un bloque de predicción de luma y a bloques correspondientes de predicción de croma. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden admitir PU que tienen diversos tamaños. Como se ha indicado anteriormente, el tamaño de una CU se puede referir al tamaño del bloque de codificación de luma de la CU, y el tamaño de una PU se puede referir al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es de $2N \times 2N$, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden admitir tamaños de PU de $2N \times 2N$ o $N \times N$ para la intrapredicción, y tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares para la interpretación. El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden admitir una partición asimétrica para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para la interpretación.

65

- 5 [0111] La unidad de procesamiento de interpredicción 120 puede generar datos predictivos para una PU realizando una interpredicción en cada PU de una CU. Los datos predictivos para la PU pueden incluir bloques predictivos de la PU e información de movimiento para la PU. La unidad de interpredicción 121 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, en un fragmento P o en un fragmento B. En un fragmento I, todas las PU se intrapredicen. Por lo tanto, si la PU está en un fragmento I, la unidad de interpredicción 121 no realiza interpredicción en la PU. Por tanto, para bloques codificados en el modo I, el bloque predicho se forma usando predicción espacial a partir de bloques vecinos previamente codificados dentro de la misma trama.
- 10 [0112] Si una PU está en un fragmento P, la unidad de estimación de movimiento de la unidad de procesamiento de interpredicción 120 puede buscar en las imágenes de referencia de una lista de imágenes de referencia (por ejemplo, "RefPicListO") una región de referencia para la PU. La región de referencia para la PU puede ser una región, dentro de una imagen de referencia, que contiene bloques de muestras que se corresponden más estrechamente con los bloques de muestras de la PU. La unidad de estimación de movimiento puede generar un índice de referencia que indica una posición en RefPicListO de la imagen de referencia que contiene la región de referencia para la PU. Además, la unidad de estimación de movimiento puede generar un MV que indica un desplazamiento espacial entre un bloque de codificación de la PU y una ubicación de referencia asociada a la región de referencia. Por ejemplo, el MV puede ser un vector bidimensional que proporciona un desplazamiento desde las coordenadas de la imagen descodificada actual hasta las coordenadas de una imagen de referencia. La unidad de estimación de movimiento puede proporcionar el índice de referencia y el MV como la información de movimiento de la PU. La unidad de compensación de movimiento de la unidad de procesamiento de interpredicción 120 puede generar los bloques predictivos de la PU basándose en muestras reales o interpoladas en la ubicación de referencia, indicada por el vector de movimiento de la PU.
- 25 [0113] Si una PU está en un fragmento B, la unidad de estimación de movimiento puede realizar unipredicción o bipredicción para la PU. Para realizar la unipredicción para la PU, la unidad de estimación de movimiento puede buscar en las imágenes de referencia de RefPicListO o de una segunda lista de imágenes de referencia ("RefPicList1") una región de referencia para la PU. La unidad de estimación de movimiento puede emitir, como información de movimiento de la PU, un índice de referencia que indica una posición en la RefPicListO o RefPicList1 de la imagen de referencia que contiene la región de referencia, un MV que indica un desplazamiento espacial entre un bloque de predicción de la PU y una ubicación de referencia asociada a la región de referencia, y uno o más indicadores de dirección de predicción que indican si la imagen de referencia está en la RefPicListO o RefPicList1. La unidad de compensación de movimiento de la unidad de procesamiento de interpredicción 120 puede generar los bloques predictivos de la PU basándose, al menos en parte, en muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.
- 35 [0114] Para realizar una interpredicción bidireccional para una PU, la unidad de estimación de movimiento puede buscar en las imágenes de referencia de la RefPicListO una región de referencia para la PU y también puede buscar en las imágenes de referencia de la RefPicList1 otra región de referencia para la PU. La unidad de estimación de movimiento puede generar índices de imágenes de referencia que indican posiciones en la RefPicListO y en la RefPicList1 de las imágenes de referencia que contienen las regiones de referencia. Además, la unidad de estimación de movimiento puede generar unos MV que indican desplazamientos espaciales entre la ubicación de referencia asociada a las regiones de referencia y un bloque de muestras de la PU. La información de movimiento de la PU puede incluir los índices de referencia y los MV de la PU. La unidad de compensación de movimiento puede generar los bloques predictivos de la PU basándose, al menos en parte, en muestras reales o interpoladas en las regiones de referencia indicadas por los vectores de movimiento de la PU.
- 50 [0115] De acuerdo con diversos ejemplos de esta divulgación, el codificador de vídeo 20 puede estar configurado para realizar una codificación basada en paletas. Con respecto al entorno de la HEVC, como ejemplo, las técnicas de codificación basada en paletas pueden configurarse para ser usadas en el nivel de CU. En otros ejemplos, las técnicas de codificación de vídeo basadas en paletas pueden configurarse para ser usadas en el nivel de PU. En otros ejemplos, las técnicas de codificación basadas en paletas pueden configurarse para ser usadas en el nivel de la unidad de predicción secundaria (sub-PU) (por ejemplo, un subbloque de una unidad de predicción). Por consiguiente, todos los procesos divulgados, descritos en el presente documento (en toda esta divulgación) en el contexto de un modo de CU, pueden, de forma adicional o alternativa, aplicarse a un nivel de PU o PU secundaria. Sin embargo, estos ejemplos basados en la HEVC no se deben considerar como una restricción o limitación de las técnicas de codificación de vídeo basada en paletas descritas en el presente documento, ya que dichas técnicas se pueden aplicar para que funcionen independientemente o como parte de otros sistemas/normas existentes o todavía por desarrollar. En estos casos, la unidad para la codificación de paletas puede ser bloques cuadrados, bloques rectangulares o incluso regiones de forma no rectangular.
- 60 [0116] La unidad de codificación basada en paletas 122, por ejemplo, puede realizar una descodificación basada en paletas cuando se selecciona un modo de codificación basada en paletas, por ejemplo, para una CU o PU. Por ejemplo, la unidad de codificación basada en paletas 122 puede configurarse para generar una paleta que tenga entradas que indiquen valores de píxeles, seleccionar valores de píxeles en una paleta para representar valores de píxeles de al menos algunas posiciones de un bloque de datos de vídeo y señalar información que asocie al

menos algunas de las posiciones del bloque de datos de vídeo con entradas en la paleta correspondientes, respectivamente, a los valores de píxeles seleccionados. Aunque se describe que la unidad de codificación basada en paletas 122 realiza diversas funciones, otras unidades de procesamiento, o una combinación de diferentes unidades de procesamiento, pueden realizar algunas o la totalidad de dichas funciones.

5 **[0117]** De acuerdo a aspectos de esta divulgación, la unidad de codificación basada en paletas 122 puede configurarse para realizar cualquier combinación de las técnicas para la codificación de paletas descritas en el presente documento.

10 **[0118]** Por ejemplo, la unidad de codificación basada en paletas 122 puede configurarse para deshabilitar la codificación del modo de paleta para cualquier bloque de paleta que tenga un tamaño de 64 x 64 o mayor. En otros ejemplos, la unidad de codificación basada en paletas 122 puede configurarse para restringir la codificación del modo de paleta a bloques de paleta que tengan un tamaño inferior a 64 x 64, lo que significa que la codificación del modo de paleta puede habilitarse o, en cualquier caso, utilizarse para bloques de paleta que tengan un tamaño inferior a 64 x 64. En otros ejemplos, la unidad de codificación basada en paletas 122 puede configurarse para restringir normativamente el tamaño de bloque de la paleta más grande basado en el tamaño de la unidad de transformada más grande. Para mencionar otro ejemplo, la unidad de codificación basada en paletas 122 puede configurarse para deshabilitar la codificación del modo de paleta para un bloque de paleta que tiene un tamaño que sobrepasa o de lo contrario es mayor que el tamaño de la unidad de transformada más grande. La unidad de codificación basada en paletas 122 puede configurarse de manera similar para realizar cualquier otra técnica para la codificación de paletas descrita en el presente documento.

20 **[0119]** La unidad de procesamiento de intrapredicción 126 puede generar datos predictivos para una PU realizando una intrapredicción en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos para la PU y diversos elementos de sintaxis. La unidad de procesamiento de intrapredicción 126 puede realizar una intrapredicción en las PU en fragmentos I, fragmentos P y fragmentos B.

25 **[0120]** Para realizar una intrapredicción en una PU, la unidad de procesamiento de intrapredicción 126 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos predictivos para la PU. La unidad de procesamiento de intrapredicción 126 puede usar muestras de bloques de muestras de PU vecinas para generar un bloque predictivo para una PU. Las PU vecinas pueden estar arriba, arriba y a la derecha, arriba y a la izquierda, o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha y de arriba abajo, para las PU, CU y CTU. La unidad de procesamiento de intrapredicción 126 puede usar diversos números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional. En algunos ejemplos, el número de modos de intrapredicción puede depender del tamaño de la región asociada a la PU.

30 **[0121]** La unidad de codificación de bloque 100 puede seleccionar los datos predictivos para las PU de una CU de entre los datos predictivos generados por la unidad de procesamiento de interpredicción 120 para las PU, o los datos predictivos generados por la unidad de procesamiento de intrapredicción 126 para las PU. En algunos ejemplos, la unidad de codificación de bloque 100 selecciona los datos predictivos para las PU de la CU basándose en métricas de velocidad/distorsión de los conjuntos de datos predictivos. Los bloques predictivos de los datos predictivos seleccionados pueden denominarse en el presente documento bloques predictivos seleccionados.

35 **[0122]** La unidad de generación residual 102 puede generar, basándose en el bloque de codificación de luma, Cb y Cr de una CU y los bloques predictivos seleccionados de luma, Cb y Cr de las PU de la CU, bloques residuales de luma, Cb y Cr de la CU. Por ejemplo, la unidad de generación residual 102 puede generar los bloques residuales de la CU de manera que cada muestra en los bloques residuales tenga un valor igual a una diferencia entre una muestra en un bloque de codificación de la CU y una muestra correspondiente en un bloque predictivo seleccionado correspondiente de una PU de la CU.

40 **[0123]** La unidad de procesamiento de transformada 104 puede realizar una partición de árbol cuaternario para dividir los bloques residuales asociados a una CU en bloques de transformada asociados a las TU de la CU. Por tanto, en algunos ejemplos, una TU puede asociarse a un bloque de transformada de luma y dos bloques de transformada de croma. Los tamaños y las posiciones de los bloques de transformada de luma y croma de las TU de una CU pueden o no estar basados en los tamaños y las posiciones de bloques de predicción de las PU de la CU. Una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados a cada una de las regiones. Las TU de una CU pueden corresponder a nodos hoja del RQT.

45 **[0124]** La unidad de procesamiento de transformada 104 puede generar bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas a un bloque de transformada asociado a una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada discreta del coseno (DCT), una transformada direccional o una transformada conceptualmente similar a un bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 104 no aplica transformadas a un bloque de transformada. En dichos ejemplos, el bloque de transformada se puede tratar como un bloque de coeficientes de transformada.

5 **[0125]** La unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de n bits puede redondearse hacia abajo hasta un coeficiente de transformada de m bits durante la cuantificación, donde n es mayor que m . La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado a una TU de una CU basándose en un valor de parámetro de cuantificación (QP) asociado a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados a una CU, ajustando el valor QP asociado a la CU. La cuantificación puede introducir pérdida de información, por lo que los coeficientes de transformada cuantificados pueden tener una precisión menor que los originales.

15 **[0126]** La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformada inversa 110 pueden aplicar una cuantificación inversa y transformadas inversas a un bloque de coeficientes, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes. La unidad de reconstrucción 112 puede añadir el bloque residual reconstruido a las muestras correspondientes de uno o más bloques predictivos generados por la unidad de codificación de bloque 100 para producir un bloque de transformada reconstruido asociado a una TU. Reconstruyendo bloques de transformada para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

20 **[0127]** La unidad de filtro 114 puede realizar una o más operaciones de desbloqueo para reducir los artefactos de los bloques en los bloques de codificación asociados a una CU. La unidad de filtro 114 puede realizar otras operaciones de filtrado, incluido el filtrado de desplazamiento adaptativo de muestras (SAO) y/o el filtrado de bucle adaptativo (ALF). La memoria intermedia de imágenes descodificadas 116 puede almacenar los bloques de codificación reconstruidos después de que la unidad de filtro 114 realice las una o más operaciones de desbloqueo en los bloques de codificación reconstruidos. La unidad de procesamiento de interpredicción 120 puede usar una imagen de referencia que contiene los bloques de codificación reconstruidos para realizar una interpredicción en las PU de otras imágenes. Además, la unidad de procesamiento de intrapredicción 126 puede usar bloques de codificación reconstruidos de la memoria intermedia de imágenes descodificadas 116 para realizar una intrapredicción en otras PU de la misma imagen que la CU.

30 **[0128]** La unidad de codificación por entropía 118 puede recibir datos desde otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 118 puede recibir bloques de coeficientes desde la unidad de cuantificación 106 y puede recibir elementos de sintaxis desde la unidad de codificación de bloque 100. La unidad de codificación por entropía 118 puede realizar una o más operaciones de codificación por entropía en los datos para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 118 puede realizar una operación de codificación adaptativa al contexto, tal como una operación CABAC, una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una operación de codificación por entropía con división de intervalos de probabilidad (PIPE), una operación de codificación exponencial-Golomb u otro tipo de operación de codificación por entropía en los datos. El codificador de vídeo 20 puede emitir un flujo de bits que incluye datos codificados por entropía, generados por la unidad de codificación por entropía 118. Por ejemplo, el flujo de bits puede incluir datos que representan un RQT para una CU.

45 **[0129]** En algunos ejemplos, la codificación residual no se realiza con la codificación de paletas. Por consiguiente, el codificador de vídeo 20 no puede realizar la transformación o la cuantificación cuando se codifica usando un modo de codificación de paletas. Además, el codificador de vídeo 20 puede codificar por entropía los datos generados usando un modo de codificación de paletas separado de los datos residuales.

50 **[0130]** De acuerdo con una o más de las técnicas de esta divulgación, el codificador de vídeo 20, y específicamente la unidad de codificación basada en paletas 122, puede realizar la codificación de vídeo basada en paletas de bloques de vídeo predichos. Como se describe anteriormente, una paleta generada por el codificador de vídeo 20 puede codificarse explícitamente y enviarse al descodificador de vídeo 30, predecirse a partir de entradas de paleta anteriores, predecirse a partir de valores de píxeles anteriores, o una combinación de los mismos.

55 **[0131]** La FIG. 3 es un diagrama de bloques que ilustra un descodificador de vídeo 30 de ejemplo que está configurado para realizar las técnicas de esta divulgación. La FIG. 3 se proporciona con propósitos explicativos y no se limita a las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con propósitos explicativos, esta divulgación describe un descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

60 **[0132]** Los detalles de la codificación de paletas descritos anteriormente con respecto al codificador 20 no se repiten aquí con respecto al descodificador 30, pero se entiende que el descodificador 30 puede realizar el proceso de descodificación recíproco en relación con cualquier proceso de codificación descrito en el presente documento con respecto al codificador 20.

65

[0133] Por ejemplo, se entiende que el descodificador de vídeo 30 puede configurarse para deshabilitar dinámicamente el modo de paleta o, en cualquier caso, configurarse para no usar el modo de paleta para ningún bloque de paleta que tenga un tamaño que no sea igual o menor que la unidad de transformada más grande. Se entiende además que el descodificador de vídeo 30 puede configurarse para descodificar un bloque de datos de vídeo usando el modo de paleta solo cuando el bloque de datos de vídeo tiene un tamaño que no sobrepasa la unidad de transformada más grande que el codificador de vídeo 20 puede configurarse para codificar y/o ese descodificador de vídeo 30 puede configurarse para descodificar. De manera similar, el descodificador de vídeo 30 puede configurarse para habilitar la codificación del modo de paleta para un bloque de datos de vídeo solo cuando el bloque de datos de vídeo tiene un tamaño que no sobrepasa la unidad de transformada más grande. En otros ejemplos, el descodificador de vídeo 30 puede configurarse para determinar si el modo de paleta está habilitado o deshabilitado en base a un valor correspondiente al indicador de modo de paleta, tal como un valor para el elemento de sintaxis de `palette_mode_flag`.

[0134] Para mencionar otro ejemplo, el descodificador de vídeo 30 puede configurarse para recibir un bloque de datos de vídeo. El descodificador de vídeo 30 puede configurarse para determinar el tamaño de los datos de vídeo en bloque en relación con el tamaño de la unidad de transformada más grande. El descodificador de vídeo 30 puede configurarse para determinar que el bloque de vídeo recibido no está codificado en modo de paleta cuando el bloque de datos de vídeo recibido es mayor que el tamaño de la unidad de transformada más grande.

[0135] El codificador de vídeo 30 representa un ejemplo de dispositivo que puede estar configurado para realizar técnicas para la codificación basada en paletas y la codificación por entropía (por ejemplo, CABAC), de acuerdo con diversos ejemplos descritos en esta divulgación.

[0136] En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 150, una memoria de datos de vídeo 151, una unidad de descodificación de bloque 152, una unidad de cuantificación inversa 154, una unidad de procesamiento de transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 160 y una memoria intermedia de imágenes descodificadas 162. La unidad de descodificación de bloque 152 incluye una unidad de compensación de movimiento 164 y una unidad de procesamiento de intrapredicción 166. El descodificador de vídeo 30 también incluye una unidad de descodificación basada en paletas 165 configurada para realizar diversos aspectos de las técnicas de codificación basada en paletas descritas en esta divulgación. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

[0137] La memoria de datos de vídeo 151 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado, a descodificar por parte de los componentes del descodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 151 pueden obtenerse, por ejemplo, a partir del medio legible por ordenador 16, por ejemplo, desde una fuente de vídeo local, tal como una cámara, mediante comunicación de datos de vídeo por red cableada o inalámbrica, o accediendo a medios de almacenamiento físico de datos. La memoria de datos de vídeo 151 puede formar una memoria intermedia de imágenes codificadas (CPB) que almacene datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. La memoria intermedia de imágenes descodificadas 162 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la descodificación de datos de vídeo mediante el descodificador de vídeo 30, por ejemplo, en los modos de intracodificación o intercodificación. La memoria de datos de vídeo 151 y la memoria intermedia de imágenes descodificadas 162 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El mismo dispositivo de memoria u otros dispositivos de memoria separados pueden proporcionar una memoria de datos de vídeo 151 y una memoria intermedia de imágenes descodificadas 162. En diversos ejemplos, la memoria de datos de vídeo 151 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera del chip con respecto a esos componentes.

[0138] Una memoria intermedia de imágenes codificadas (CPB) puede recibir y almacenar datos de vídeo codificados (por ejemplo, unidades NAL) de un flujo de bits. La unidad de descodificación por entropía 150 puede recibir datos de vídeo codificados (por ejemplo, unidades NAL) desde la CPB y analizar las unidades NAL para descodificar elementos de sintaxis. La unidad de descodificación por entropía 150 puede descodificar por entropía elementos de sintaxis codificados por entropía en las unidades NAL. La unidad de descodificación de bloque 152, la unidad de cuantificación inversa 154, la unidad de procesamiento de transformada inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 160 pueden generar datos de vídeo descodificados basándose en los elementos de sintaxis extraídos del flujo de bits.

[0139] El descodificador de vídeo 30 puede configurarse para realizar un proceso en general recíproco al del codificador de vídeo 20 descrito en el presente documento. De manera similar, el codificador de vídeo 20 puede configurarse para realizar un proceso en general recíproco al del descodificador de vídeo 20 descrito en el presente documento. Por ejemplo, la divulgación de que el descodificador de vídeo 30 puede configurarse para descodificar un elemento de sintaxis codificada en un flujo de bits del mismo modo necesariamente divulga que el codificador de vídeo 20 puede configurarse para codificar el elemento de sintaxis en el flujo de bits.

[0140] Para mencionar otro ejemplo, la unidad de descodificación por entropía 150 puede configurarse para realizar un proceso en general recíproco al de la unidad de codificación por entropía 118 descrita en el presente documento. De acuerdo con aspectos de esta divulgación, la unidad de descodificación por entropía 150 puede configurarse para descodificar por entropía cualquier palabra de código generada por la unidad de codificación por entropía 118.

[0141] Las unidades NAL del flujo de bits pueden incluir unidades NAL de fragmentos codificados. Como parte de la descodificación del flujo de bits, la unidad de descodificación por entropía 150 puede extraer y descodificar por entropía elementos de sintaxis de las unidades NAL de fragmentos codificados. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos de sintaxis pertenecientes a un fragmento. Los elementos de sintaxis de la cabecera de fragmento pueden incluir un elemento de sintaxis que identifica un PPS asociado a una imagen que contiene el fragmento.

[0142] Además de descodificar elementos de sintaxis del flujo de bits, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Realizando la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir bloques residuales de la CU.

[0143] Como parte de realizar una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede realizar la cuantificación inversa, es decir, descuantificar, los bloques de coeficientes asociados a la TU. La unidad de cuantificación inversa 154 puede usar un valor QP asociado a la CU de la TU para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para que la unidad de cuantificación inversa 154 lo aplique. Es decir, la relación de compresión, es decir, la relación entre el número de bits usados para representar la secuencia original y la comprimida, puede controlarse ajustando el valor del QP usado al cuantificar los coeficientes de transformada. La relación de compresión también puede depender del procedimiento de codificación por entropía empleado.

[0144] Después de que la unidad de cuantificación inversa 154 haya realizado la cuantificación inversa de un bloque de coeficientes, la unidad de procesamiento de transformada inversa 156 puede aplicar una o más transformadas inversas al bloque de coeficientes con objeto de generar un bloque residual asociado a la TU. Por ejemplo, la unidad de procesamiento de transformada inversa 156 puede aplicar una DCT inversa, una transformada entera inversa, una transformada inversa de Karhunen-Loeve (KLT), una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes.

[0145] Si se codifica una PU usando intrapredicción, la unidad de procesamiento de intrapredicción 166 puede realizar una intrapredicción para generar bloques predictivos para la PU. La unidad de procesamiento de intrapredicción 166 puede usar un modo de intrapredicción para generar los bloques predictivos de luma, Cb y Cr para la PU basándose en los bloques de predicción de las PU espacialmente vecinas. La unidad de procesamiento de intrapredicción 166 puede determinar el modo de intrapredicción para la PU basándose en uno o más elementos de sintaxis descodificados a partir del flujo de bits.

[0146] La unidad de descodificación de bloque 152 puede construir una primera lista de imágenes de referencia (RefPicList0) y una segunda lista de imágenes de referencia (RefPicList1) basándose en elementos de sintaxis extraídos del flujo de bits. Además, si una PU se codifica usando interpredicción, la unidad de descodificación por entropía 150 puede extraer información de movimiento para la PU. La unidad de compensación de movimiento 164 puede determinar, basándose en la información de movimiento de la PU, una o más regiones de referencia para la PU. La unidad de compensación de movimiento 164 puede generar, basándose en bloques de muestras de los uno o más bloques de referencia para la PU, bloques predictivos de luma, Cb y Cr para la PU.

[0147] La unidad de reconstrucción 158 puede usar los bloques de transformada de luma, Cb y Cr, asociados a las TU de una CU y los bloques predictivos de luma, Cb y Cr de las PU de la CU, es decir, datos de intrapredicción o datos de interpredicción, según corresponda, para reconstruir los bloques de codificación de luma, Cb y Cr de la CU. Por ejemplo, la unidad de reconstrucción 158 puede añadir muestras de los bloques de transformada de luma, Cb y Cr a muestras correspondientes de los bloques predictivos de luma, Cb y Cr para reconstruir los bloques de codificación de luma, Cb y Cr de la CU.

[0148] La unidad de filtro 160 puede realizar una operación de desbloqueo para reducir los artefactos de bloque asociados a los bloques de codificación de luma, Cb y Cr de la CU. El descodificador de vídeo 30 puede almacenar los bloques de codificación de luma, Cb y Cr de la CU en la memoria intermedia de imágenes descodificadas 162. La memoria intermedia de imágenes descodificadas 162 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intrapredicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en

los bloques de luma, Cb y Cr de la memoria intermedia de imágenes descodificadas 162, operaciones de intrapredicción o de interpredicción en las PU de otras CU.

5 **[0149]** De acuerdo con diversos ejemplos de esta divulgación, el descodificador de vídeo 30 puede estar configurado para realizar una codificación basada en paletas. La unidad de descodificación basada en paletas 165, por ejemplo, puede realizar una descodificación basada en paletas cuando se selecciona un modo de descodificación basada en paletas, por ejemplo, para una CU o PU. Por ejemplo, la unidad de descodificación basada en paletas 165 puede estar configurada para generar una paleta que tenga entradas que indiquen valores de píxeles, recibir información que asocie al menos algunas ubicaciones de píxel en un bloque de datos de vídeo
10 con entradas de la paleta, seleccionar valores de píxeles de la paleta en base a la información y reconstruir valores de píxeles del bloque en base a los valores de píxeles seleccionados de la paleta. Aunque se describe que la unidad de descodificación basada en paletas 165 realiza diversas funciones, otras unidades de procesamiento o una combinación de diferentes unidades de procesamiento pueden realizar algunas o la totalidad de dichas funciones.

15 **[0150]** La unidad de descodificación basada en paletas 165 puede recibir información de modo de codificación de paletas, y realizar las operaciones anteriores cuando la información del modo de codificación de paletas indica que el modo de codificación de paletas se aplica al bloque. Cuando la información del modo de codificación de paletas indica que el modo de codificación de paletas no se aplica al bloque, o cuando otra información de modo indica el uso de un modo diferente, la unidad de descodificación basada en paletas 165 descodifica el bloque de datos de vídeo utilizando un modo de codificación basada en paletas, por ejemplo, tal como un modo de codificación HEVC interpredictivo o intrapredictivo. El bloque de datos de vídeo puede ser, por ejemplo, una CU o PU generada de acuerdo con un proceso de codificación HEVC. El modo de codificación basada en paletas puede comprender uno de una pluralidad de diferentes modos de codificación basada en paletas, o puede haber un único modo de codificación basada en paletas.
20

25 **[0151]** De acuerdo a aspectos de esta divulgación, la unidad de descodificación basada en paletas 165 puede configurarse para realizar cualquier combinación de las técnicas para la codificación de paletas descritas en el presente documento. Los detalles de la codificación de paletas descritos anteriormente con respecto al codificador 20 no se repiten aquí con respecto al descodificador 30, pero se entiende que el descodificador 30 puede realizar el proceso de descodificación recíproco basado en paletas en relación con cualquier proceso de codificación basado en paletas descrito en el presente documento con respecto al codificador 20.
30

35 **[0152]** La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de determinación de una paleta para codificar datos de vídeo, congruente con las técnicas de esta divulgación. El ejemplo de la FIG. 4 incluye una imagen 178 que tiene una primera unidad de codificación (CU) de PAL (paleta) 180 que está asociada a unas primeras paletas 184 y una segunda CU de PAL 188 que está asociada a unas segundas paletas 192. Como se describe más en detalle a continuación y de acuerdo con las técnicas de esta divulgación, las segundas paletas 192 están basadas en las primeras paletas 184. La imagen 178 también incluye el bloque 196 codificado con un modo de codificación de intrapredicción y el bloque 200 que está codificado con un modo de codificación de interpredicción.
40

45 **[0153]** Las técnicas de la FIG. 4 se describen en el contexto del codificador de vídeo 20 (FIG. 1 y FIG. 2) y el descodificador de vídeo 30 (FIG. 1 y FIG. 3) y con respecto a la norma de codificación de vídeo HEVC, con propósitos explicativos. Sin embargo, se debería entender que las técnicas de esta divulgación no están limitadas de esta manera, y que otros procesadores y/o dispositivos de codificación de vídeo pueden aplicarlas en otros procesos y/o normas de codificación de vídeo.

50 **[0154]** En general, una paleta se refiere a un número de valores de píxeles que son dominantes y/o representativos para una CU actualmente codificada, la CU 188 en el ejemplo de la FIG. 4. Las primeras paletas 184 (que también pueden denominarse índices 184) y las segundas paletas 192 (que también pueden denominarse índices 192) se muestran como que incluyen múltiples paletas (que también pueden denominarse índices múltiples). En algunos ejemplos, de acuerdo con aspectos de esta divulgación, un codificador de vídeo (tal como el codificador de vídeo 20 o el descodificador de vídeo 30) puede codificar paletas (por ejemplo, índices) por separado para cada componente de color de una CU. Por ejemplo, el codificador de vídeo 20 puede codificar una paleta para un componente de luma (Y) de una CU, otra paleta para un componente de croma (U) de la CU y otra paleta más para el componente de croma (V) de la CU. En este ejemplo, unas entradas de la paleta Y pueden representar valores Y de píxeles de la CU, unas entradas de la paleta U pueden representar valores U de píxeles de la CU y unas entradas de la paleta V pueden representar valores V de píxeles de la CU.
55

60 **[0155]** En otros ejemplos, el codificador de vídeo 20 puede codificar una única paleta para todos los componentes de color de una CU. En este ejemplo, el codificador de vídeo 20 puede codificar una paleta que tiene una i-ésima entrada que es un valor triple, que incluye Y_i , U_i y V_i . En este caso, la paleta incluye valores para cada uno de los componentes de los píxeles. Por consiguiente, la representación de las paletas 184 y 192 como un conjunto de paletas que tiene múltiples paletas individuales es simplemente un ejemplo y no pretende ser limitante.

65 **[0156]** En el ejemplo de la FIG. 4, la primera paleta 184 incluye tres entradas 202-206 que tienen un valor del índice de entrada 1, un valor del índice de entrada 2 y un valor del índice de entrada 3, respectivamente. Las

primeras paletas 184 relacionan los valores de índices (por ejemplo, los valores que se muestran en la columna izquierda de las primeras paletas 184) con los valores de píxeles. Por ejemplo, como se muestra en la FIG. 4, una de las primeras paletas 184 relaciona los valores de índices 1, 2 y 3 con los valores de píxeles A, B y C, respectivamente. Como se describe en el presente documento, en vez de codificar los valores de píxeles reales de la primera CU 180, un codificador de vídeo (tal como un codificador de vídeo 20 o un descodificador de vídeo 30) puede usar la codificación basada en paletas para codificar los píxeles del bloque usando los índices 1-3 (que también se puede expresar como valores de índices 1-3). Es decir, para cada posición de píxel de la primera CU 180, el codificador de vídeo 20 puede codificar un valor del índice para el píxel, donde el valor del índice está asociado a un valor de píxel en una o más de las primeras paletas 184. El descodificador de vídeo 30 puede obtener los valores de índices a partir de un flujo de bits y reconstruir los valores de píxeles usando los valores de índices y una o más de las primeras paletas 184. Por tanto, el codificador de vídeo 20 transmite unas primeras paletas 184 en un flujo de bits de datos de vídeo codificado para su uso por el descodificador de vídeo 30 en la descodificación basada en paletas.

[0157] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden determinar unas segundas paletas 192 basadas en las primeras paletas 184. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden ubicar uno o más bloques a partir de los cuales se determinan las paletas predictivas, en este ejemplo, las primeras paletas 184. En algunos ejemplos, tales como el ejemplo que se ilustra en la FIG. 4, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden ubicar la CU previamente codificada como una CU izquierda vecina (primera CU 180) al determinar una paleta predictiva para la segunda CU 188.

[0158] En el ejemplo de la FIG. 4, las segundas paletas 192 incluyen tres entradas 208-212 que tienen un valor del índice de entrada 1, un valor del índice de entrada 2 y un valor del índice de entrada 3, respectivamente. Las segundas paletas 192 relacionan los valores de índices (por ejemplo, los valores que se muestran en la columna izquierda de las primeras paletas 192) con los valores de píxeles. Por ejemplo, como se muestra en la FIG. 4, una de las segundas paletas 192 relaciona los valores de índices 1, 2 y 3 con los valores de píxeles A, B y D, respectivamente. En este ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos de sintaxis que indican qué entradas de las primeras paletas 184 están incluidas en las segundas paletas 192. En el ejemplo de la FIG. 4, los uno o más elementos de sintaxis se ilustran como un vector 216. El vector 216 tiene un número de bins (o bits) asociados, con cada bin que indica si el predictor de la paleta asociado a ese bin se usa para predecir una entrada de la paleta actual. Por ejemplo, el vector 216 indica que las dos primeras entradas de las primeras paletas 184 (202 y 204) están incluidas en las segundas paletas 192 (un valor de "1" en el vector 216), mientras que la tercera entrada de las primeras paletas 184 no está incluida en las segundas paletas 192 (un valor de "0" en el vector 216). En el ejemplo de la FIG. 4, el vector es un vector booleano.

[0159] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden determinar una lista de predictores de paleta (que también puede denominarse tabla de predictores de paleta) cuando se realiza la predicción de la paleta. La lista de predictores de paleta puede incluir entradas de paletas de uno o más bloques vecinos que se usan para predecir una o más entradas de una paleta para codificar un bloque actual. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden construir la lista de la misma manera. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar datos (tales como el vector 216) para indicar qué entradas de la lista de predictores de paleta se tienen que incluir en una paleta para codificar un bloque actual.

[0160] La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de determinación de índices en una paleta para un bloque de píxeles, congruente con las técnicas de esta divulgación. Por ejemplo, la FIG. 5 incluye un bloque de índices 240 (que también puede denominarse mapa 240 o mapa de índices 240) que incluye valores de índices (por ejemplo, valores de índices 1, 2 y 3) que se relacionan con las posiciones respectivas de píxeles asociadas a los valores de índices de una entrada de las paletas 244.

[0161] Aunque el mapa 240 que se ilustra en el ejemplo de la FIG. 5 incluye un valor del índice para cada posición de píxel, se debería entender que, en otros ejemplos, no todas las posiciones de píxel pueden estar asociadas a un valor del índice que relaciona el valor del píxel con una entrada de las paletas 244. Es decir, como se indica anteriormente, en algunos ejemplos, el codificador de vídeo 20 puede codificar (y el descodificador de vídeo 30 puede obtener, a partir de un flujo de bits codificado) una indicación de un valor de píxel real (o su versión cuantificada) para una posición en el bloque de índices 240 si el valor del píxel no está incluido en las paletas 244.

[0162] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar configurados para codificar un mapa adicional que indica qué posiciones de píxel están asociadas a valores de índices. Por ejemplo, se va a suponer que la entrada (i, j) del bloque de índices 240 corresponde a la posición (i, j) de una CU. El codificador de vídeo 20 puede codificar uno o más elementos de sintaxis para cada entrada del bloque de índices (es decir, cada posición de píxel) que indican si la entrada tiene un valor del índice asociado. Por ejemplo, el codificador de vídeo 20 puede codificar un indicador que tiene un valor de uno para indicar que el valor del píxel en la ubicación (i, j) de la CU es uno de los valores de las paletas 244.

[0163] El codificador de vídeo 20 puede, en dicho ejemplo, codificar también una paleta (se muestra en el ejemplo de la FIG. 5 como 244). En casos en los que las paletas 244 incluyen una sola entrada y un valor de píxel asociado,

el codificador de vídeo 20 puede omitir la señalización del valor del índice. El codificador de vídeo 20 puede codificar el indicador para que tenga un valor de cero para indicar que el valor del píxel en la ubicación (i, j) de la CU no es uno de los valores de las paletas 244. En este ejemplo, el codificador de vídeo 20 también puede codificar una indicación del valor del píxel para su uso por el descodificador de vídeo 30 al reconstruir el valor del píxel. En algunos casos, el valor del píxel se puede codificar de una manera que conlleva pérdidas.

[0164] El valor de un píxel en una posición de una CU puede proporcionar una indicación de valores de uno o más píxeles diferentes en otras posiciones de la CU. Por ejemplo, puede haber una probabilidad relativamente alta de que unas posiciones de píxeles vecinos de una CU tengan el mismo valor de píxel o estén correlacionadas con el mismo valor del índice (en el caso de codificación con pérdidas, en la que más de un valor de píxel puede estar correlacionado con un único valor del índice).

[0165] Por consiguiente, el codificador de vídeo 20 puede codificar uno o más elementos de sintaxis que indican un número de píxeles o valores de índices consecutivos en un orden de barrido dado que tienen el mismo valor de píxel o valor del índice. Como se indica anteriormente, la cadena de valores de píxeles o índice con igual valor se puede denominar ejecución en el presente documento. En un ejemplo con propósitos ilustrativos, si dos píxeles o índices consecutivos en un orden de barrido dado tienen valores diferentes, la ejecución es igual a cero. Si dos píxeles o índices consecutivos en un orden de barrido dado tienen el mismo valor, pero el tercer píxel o índice en el orden de barrido tiene un valor diferente, la ejecución es igual a uno. Para tres índices o píxeles consecutivos con el mismo valor, la ejecución es dos, y así sucesivamente. El descodificador de vídeo 30 puede obtener los elementos de sintaxis que indican una ejecución a partir de un flujo de bits codificado, y usar los datos para determinar el número de ubicaciones consecutivas que tienen el mismo valor de píxel o índice.

[0166] En algunos ejemplos de acuerdo con las técnicas de esta divulgación, la unidad de codificación por entropía 118 y la unidad de descodificación por entropía 150 pueden configurarse para codificar por entropía el bloque de índices 240. Por ejemplo, la unidad de codificación 118 y la unidad de descodificación por entropía 150 se pueden configurar para codificar por entropía longitudes de ejecución (por ejemplo, valores o códigos de longitud de ejecución) y/o un vector de predicción de paleta binario relacionado con un bloque de índices en el modo de paleta.

[0167] La FIG. 6 es un diagrama conceptual que ilustra un ejemplo de determinación de una longitud de ejecución máxima con copia arriba, suponiendo un ejemplo de un orden de barrido de trama, congruente con las técnicas de esta divulgación. En el ejemplo de la FIG. 6, si ninguno de los píxeles que engloban las líneas discontinuas 280 se codifica como una muestra de escape, la longitud de ejecución máxima posible es 35 (es decir, el número de posiciones de píxeles sin sombrear). Si uno o más de los píxeles dentro de las líneas discontinuas 280 se codifican como una muestra de escape, suponiendo que el píxel marcado como el píxel de escape (la posición del píxel con la "X") es el primer píxel de escape dentro de las líneas discontinuas 280 en el orden de barrido, entonces la máxima longitud de ejecución posible codificada con copia arriba es cinco.

[0168] En algunos ejemplos, el descodificador de vídeo 30 solo puede determinar el modo de ejecución (por ejemplo, el modo de paleta en el que se codifican los píxeles) para los píxeles dentro de las líneas discontinuas 280. Por lo tanto, en el peor de los casos, el descodificador de vídeo 30 determina AnchoBloque-1 píxeles. En algunos ejemplos, el descodificador de vídeo 30 puede configurarse para implementar ciertas restricciones con respecto al número máximo de píxeles para los cuales se verifica el modo de ejecución. Por ejemplo, el descodificador de vídeo 30 solo puede verificar los píxeles dentro de las líneas discontinuas 280 si los píxeles están en la misma fila que el píxel actual. El descodificador de vídeo 30 puede inferir que todos los demás píxeles dentro de las líneas discontinuas 280 no están codificados como muestras de escape. El ejemplo en la FIG. 6 supone un orden de barrido de trama. Sin embargo, las técnicas se pueden aplicar a otros órdenes de barrido, como el recorrido horizontal y recorrido vertical.

[0169] La FIG. 7 es un diagrama de flujo que ilustra un proceso de ejemplo para la codificación de datos de vídeo congruente con las técnicas de esta divulgación. El proceso de la FIG. 7 se describe en general como que se realiza por un codificador de vídeo (por ejemplo, el codificador de vídeo 20) con fines ilustrativos, aunque una variedad de otros procesadores también puede llevar a cabo el proceso que se muestra en la FIG. 7. En algunos ejemplos, la unidad de codificación de bloque 100, la unidad de codificación basada en paletas 122 y/o la unidad de codificación por entropía 118 pueden realizar uno o más procesos que se muestran en la FIG. 7.

[0170] En el ejemplo de la FIG. 7, un codificador de vídeo (por ejemplo, codificador de vídeo 20) puede configurarse para recibir un bloque de datos de vídeo que tiene un tamaño (700). El codificador de vídeo puede configurarse para determinar el tamaño del bloque de datos de vídeo (702). El codificador de vídeo puede configurarse para deshabilitar la codificación del modo de paleta para el bloque de datos de vídeo basado en el tamaño determinado del bloque de datos de vídeo (704).

[0171] En algunos ejemplos, el codificador de vídeo puede configurarse para restringir la codificación del modo de paleta a cualquier bloque de datos de vídeo que tenga un primer tamaño menor que un segundo tamaño. En este ejemplo, el primer tamaño puede ser 32x32. En algunos ejemplos, el segundo tamaño puede ser 64 x 64. En dichos ejemplos, el codificador de vídeo puede configurarse para restringir el modo de paleta a cualquier bloque de datos

de vídeo que tenga un primer tamaño inferior a 64 x 64. En algunos casos, el primer tamaño puede comprender 32x32 y el segundo tamaño puede comprender 64x64.

5 **[0172]** En algunos ejemplos, el codificador de vídeo puede configurarse para restringir la codificación del modo de paleta a cualquier bloque de datos de vídeo que tenga un primer tamaño menor o igual al tamaño de la unidad de transformada más grande especificada para los datos de vídeo. El tamaño de la unidad de transformada más grande puede ser 32 x 32. En dicho ejemplo, el codificador de vídeo puede configurarse para restringir la codificación del modo de paleta a cualquier bloque de datos de vídeo que tenga un primer tamaño menor o igual a 32 x 32.

10 **[0173]** En algunos ejemplos, el codificador de vídeo puede configurarse para dividir el bloque de datos de vídeo en una pluralidad de 4 x 4 subbloques. En dichos ejemplos, el codificador de vídeo puede configurarse para codificar, usando el modo de paleta, la pluralidad de 4 x 4 subbloques.

15 **[0174]** En algunos ejemplos, el codificador de vídeo puede configurarse para restringir cualquier valor de longitud de ejecución en la codificación del modo de paleta a un valor de longitud de ejecución máxima de manera que la codificación del modo de paleta esté deshabilitada para el bloque de datos de vídeo en base al tamaño determinado del bloque de datos de vídeo solo si algún valor de longitud de ejecución en la codificación del modo de paleta no está restringido al valor de longitud de ejecución máxima. En un ejemplo, el valor de longitud de ejecución máxima es 32x32 menos 1. En otro ejemplo, la longitud de ejecución máxima se basa en el tamaño de la unidad de transformada más grande. En este ejemplo, si el tamaño de la unidad de transformada más grande es 32x32, entonces la longitud de ejecución máxima puede ser inferior a 32x32, tal como 32x32 menos 1. En otro ejemplo, la longitud de ejecución máxima se basa en un número de coeficientes de la unidad de transformada más grande.

25 **[0175]** Debería entenderse que todas las técnicas descritas en el presente documento pueden usarse individualmente o en combinación. Por ejemplo, el codificador de vídeo 20 y/o uno o más componentes del mismo y el descodificador de vídeo 30 y/o uno o más componentes del mismo pueden realizar las técnicas descritas en esta divulgación en cualquier combinación.

30 **[0176]** Se debe reconocer que, dependiendo del ejemplo, determinadas acciones o eventos de cualquiera de las técnicas descritas en el presente documento se pueden realizar en una secuencia distinta, se pueden añadir, fusionar o excluir por completo (por ejemplo, no todas las acciones o eventos descritos son necesarios para la puesta en práctica de las técnicas). Por otro lado, en determinados ejemplos, las acciones o eventos pueden tener lugar simultáneamente, por ejemplo, a través de procesamiento de múltiples hilos de ejecución, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente. Además, aunque con propósitos de claridad se describe que un único módulo o unidad realiza determinados aspectos de esta divulgación, se debería entender que una combinación de unidades o módulos asociados a un codificador de vídeo pueden realizar las técnicas de esta divulgación.

40 **[0177]** Determinados aspectos de esta divulgación se han descrito con respecto a la norma de HEVC en desarrollo con fines ilustrativos. Sin embargo, las técnicas descritas en esta divulgación pueden ser útiles para otros procesos de codificación de vídeo, que incluyen otros procesos de codificación de vídeo, estándar o de propiedad, aún no desarrollados.

45 **[0178]** El codificador de vídeo 20 (FIGS. 1 y 2) y/o el descodificador de vídeo 30 (FIGS. 1 y 3), los cuales se pueden denominar en general codificadores de vídeo, pueden realizar las técnicas descritas anteriormente. Asimismo, la codificación de vídeo se puede referir a una codificación de vídeo o una descodificación de vídeo, según corresponda.

50 **[0179]** De acuerdo con esta divulgación, el término "o" puede interrumpirse como "y/o" donde el contexto no dicte lo contrario. Además, aunque frases como "uno o más" o "al menos uno" o similares pueden haber sido utilizadas para algunas características divulgadas en el presente documento pero no para otras; las características para las cuales no se ha usado dicho lenguaje pueden interpretarse como que tienen un significado implícito donde el contexto no dicte lo contrario.

55 **[0180]** Aunque se han descrito anteriormente combinaciones particulares de diversos aspectos de las técnicas, estas combinaciones se proporcionan simplemente para ilustrar ejemplos de las técnicas descritas en esta divulgación. Por consiguiente, las técnicas de esta divulgación no se deben limitar a estas combinaciones de ejemplos y pueden abarcar cualquier combinación concebible de los diversos aspectos de las técnicas descritas en esta divulgación, en la medida en que dicha combinación se encuentre dentro del alcance de las reivindicaciones adjuntas.

60 **[0181]** En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en programa informático, las funciones se pueden almacenar en o transmitir a través de un medio legible por ordenador como una o más instrucciones o código, 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 unos medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que son no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0182] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda usar para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debe entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se refieren a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, donde algunos discos normalmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por ordenador.

[0183] 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 específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA) u otra circuitería lógica integrada o discreta equivalente. Por consiguiente, el término "procesador", como se usa en el presente documento, se puede referir a cualquier estructura anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar en módulos de hardware y/o de software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

[0184] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores como los descritos anteriormente, junto con software y/o firmware adecuado.

[0185] En el presente documento se han descrito diversos ejemplos. Cualquier combinación de los sistemas, operaciones, funciones o ejemplos descritos se contempla en la medida en que dicha combinación se encuentre dentro del alcance de las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un procedimiento de codificación de datos de vídeo, el procedimiento que comprende:

5 recibir un bloque de datos de vídeo que tiene un tamaño (700);
 determinar el tamaño del bloque de datos de vídeo (702); y
10 para un valor de longitud de ejecución máxima que es menor que el tamaño de la unidad de transformada más grande o que es 32×32 menos 1, uno de:
 restringir cualquier valor de longitud de ejecución para el bloque de datos de vídeo al valor de longitud de ejecución máxima cuando el modo de paleta codifica el bloque de datos de vídeo; o
15 deshabilitar, si cualquier valor de longitud de ejecución para el bloque de datos de vídeo no está restringido al valor de longitud de ejecución máxima, la codificación del modo de paleta para el bloque de datos de vídeo cuando el tamaño determinado del bloque de datos de vídeo es mayor que el tamaño de la unidad de transformada más grande (704).

20 2. El procedimiento de la reivindicación 1, en el que el tamaño de la unidad de transformada más grande es 32×32 .

3. Un aparato para codificar datos de vídeo, el aparato que comprende:

25 medios para recibir un bloque de datos de vídeo que tiene un tamaño;
 medios para determinar el tamaño del bloque de datos de vídeo; y
 medios para, para un valor de longitud de ejecución máxima que es menor que el tamaño de la unidad de transformada más grande o que es 32×32 menos 1, uno de:
30 restringir cualquier valor de longitud de ejecución para el bloque de datos de vídeo al valor de longitud de ejecución máxima cuando el modo de paleta codifica el bloque de datos de vídeo; o
 deshabilitar, si cualquier valor de longitud de ejecución para el bloque de datos de vídeo no está restringido al valor de longitud de ejecución máxima, la codificación del modo de paleta para el bloque de datos de vídeo cuando el tamaño determinado del bloque de datos de vídeo es mayor que el tamaño de la unidad de transformada más grande.
35 4. El aparato de la reivindicación 3, en el que el tamaño de la unidad de transformada más grande es 32×32 .

40 5. Un medio de almacenamiento legible por ordenador no transitorio que tiene instrucciones almacenadas en el mismo que, al ejecutarse, hacen que uno o más procesadores:

45 reciban un bloque de datos de vídeo que tengan un tamaño;
 determinen el tamaño del bloque de datos de vídeo; y
 para un valor de longitud de ejecución máxima que es menor que el tamaño de la unidad de transformada más grande o que es 32×32 menos 1, uno de:
50 restringir cualquier valor de longitud de ejecución para el bloque de datos de vídeo al valor de longitud de ejecución máxima cuando el modo de paleta codifica el bloque de datos de vídeo; o
 deshabilitar, si cualquier valor de longitud de ejecución para el bloque de datos de vídeo no está restringido al valor de longitud de ejecución máxima, la codificación del modo de paleta para el bloque de datos de vídeo cuando el tamaño determinado del bloque de datos de vídeo es mayor que el tamaño de la unidad de transformada más grande.
55 5. Un medio de almacenamiento legible por ordenador no transitorio que tiene instrucciones almacenadas en el mismo que, al ejecutarse, hacen que uno o más procesadores:

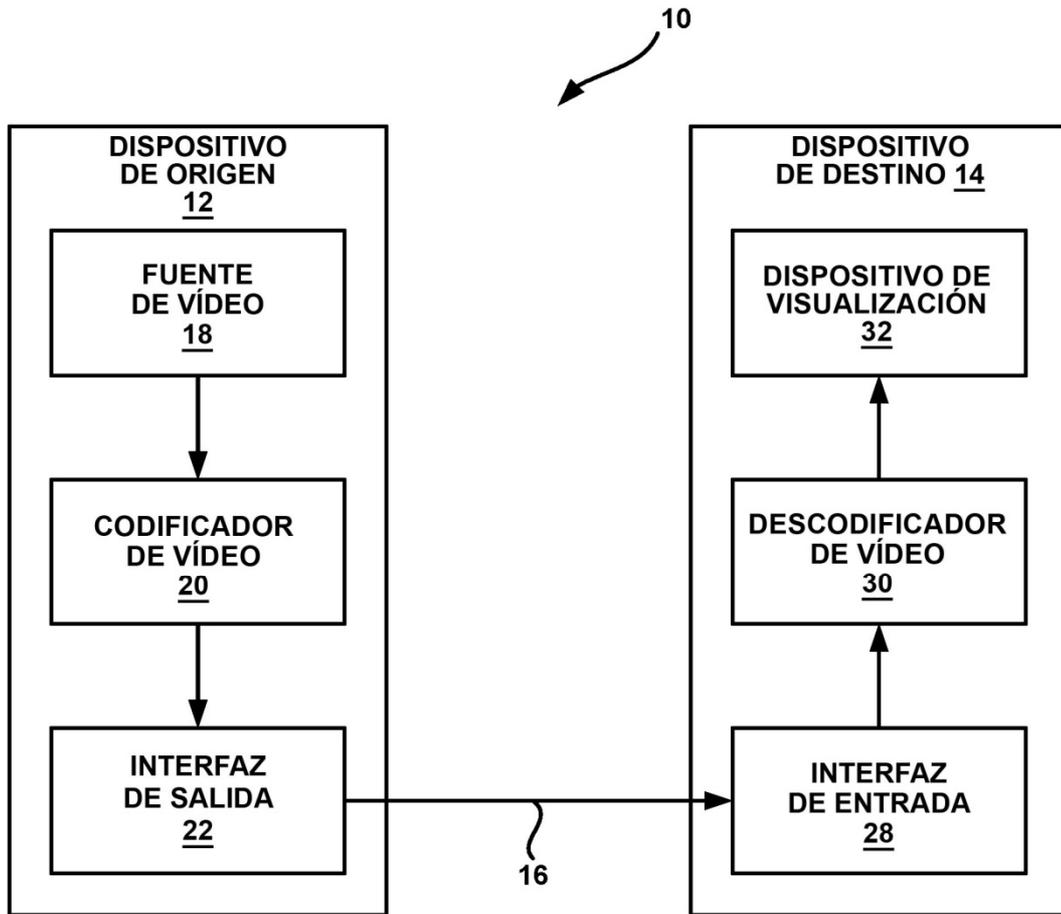


FIG. 1

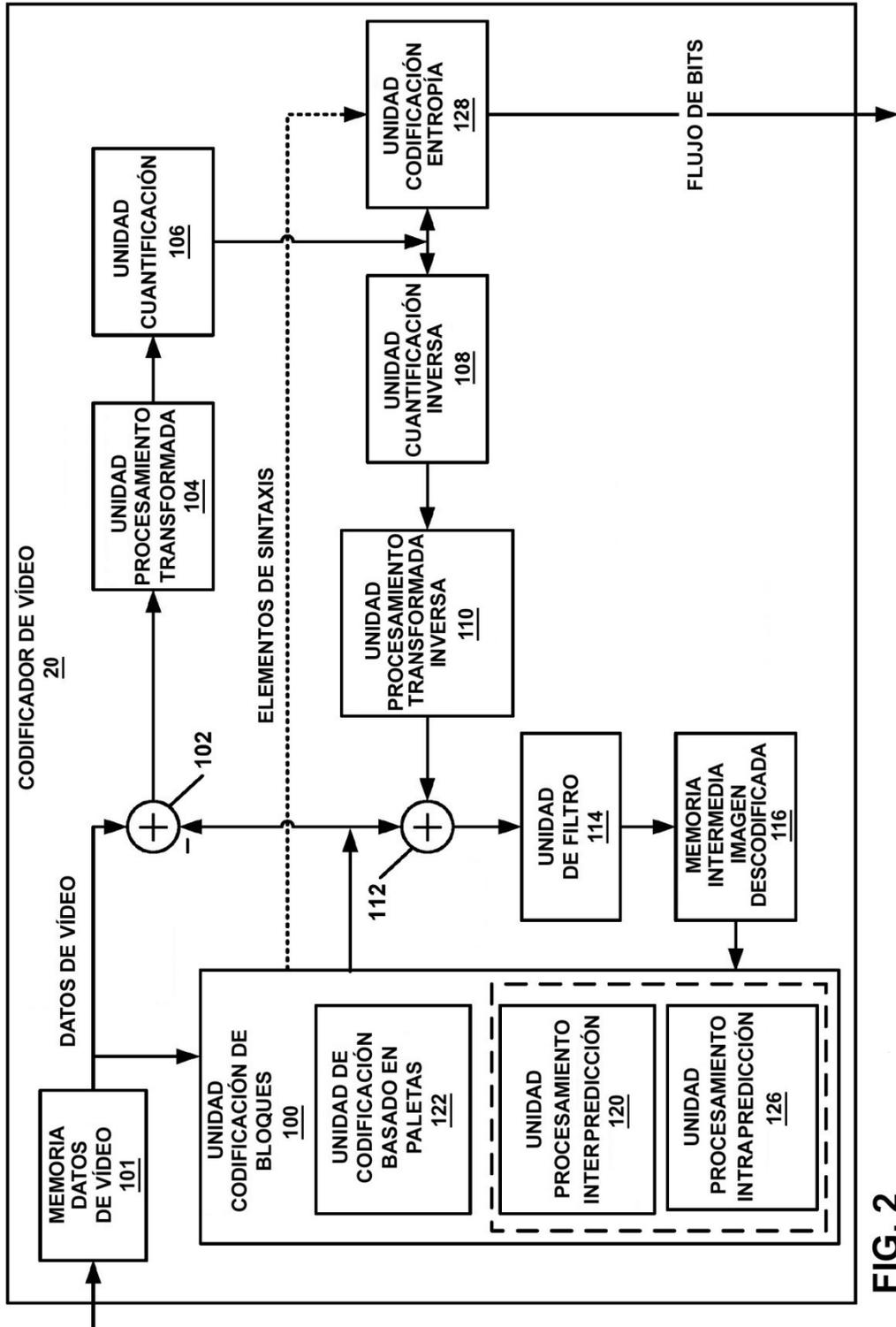


FIG. 2

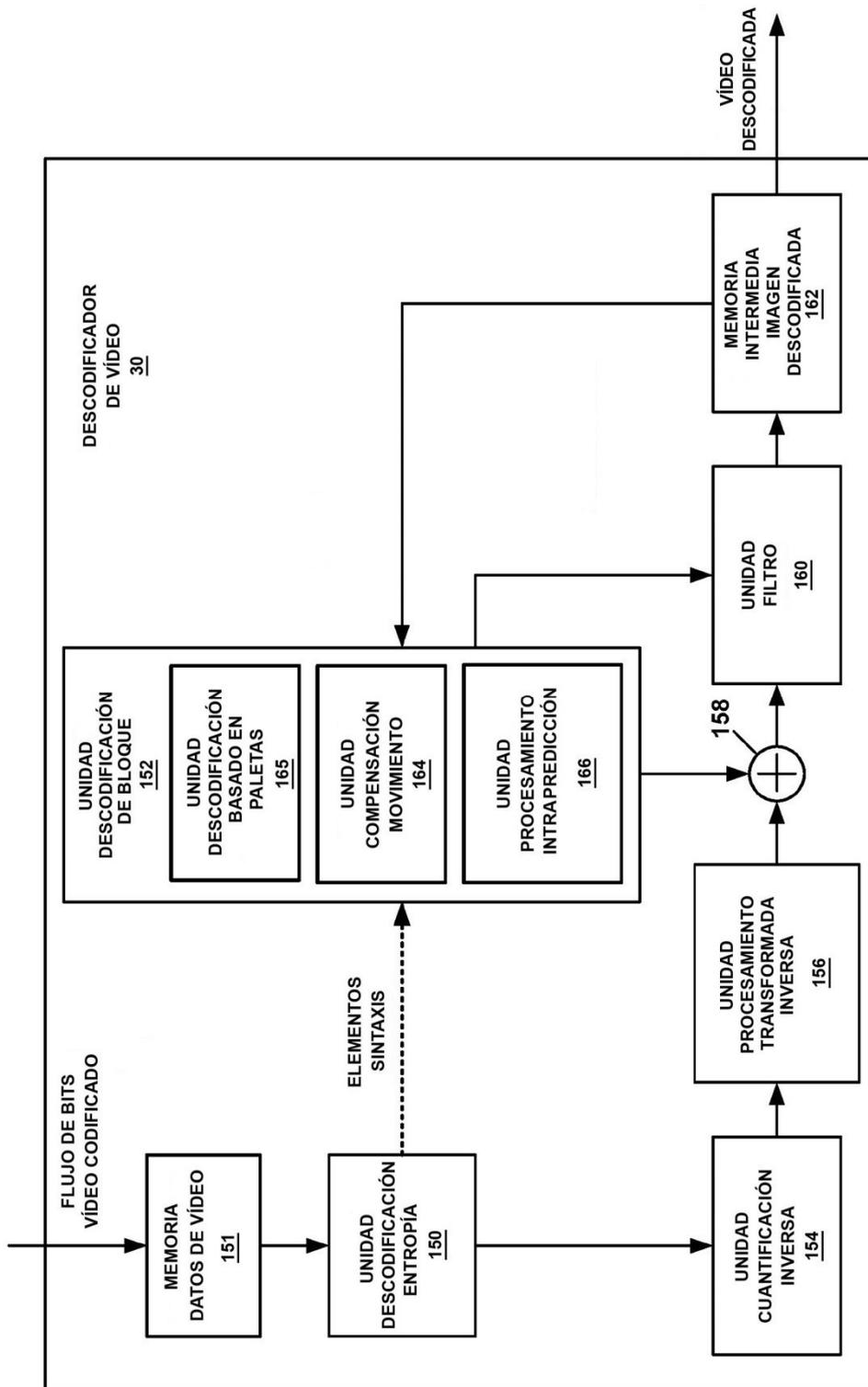


FIG. 3

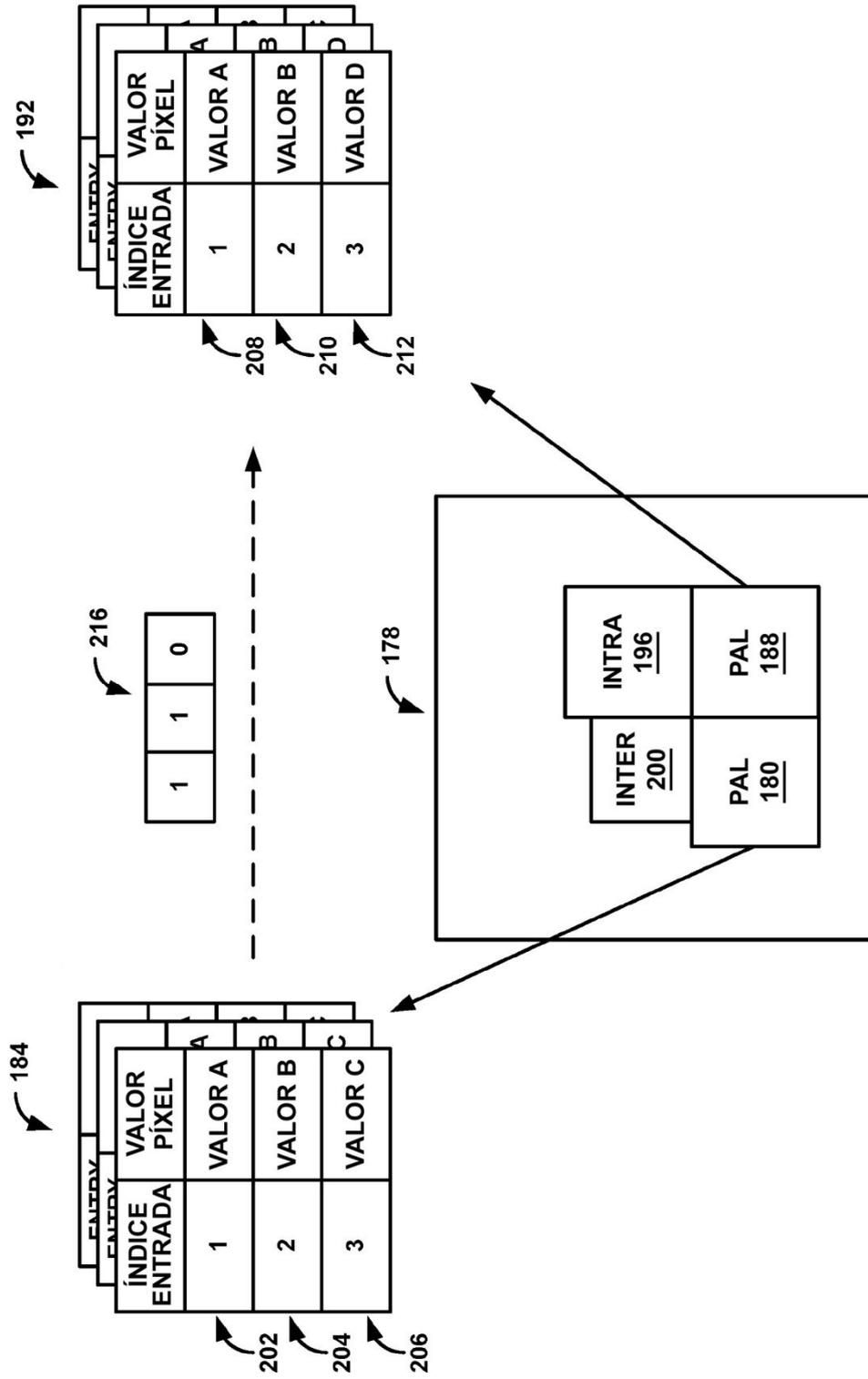


FIG. 4

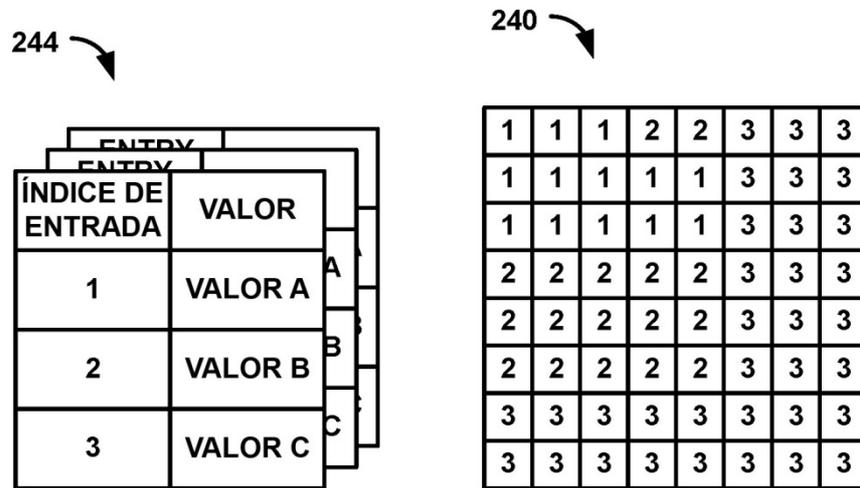


FIG. 5

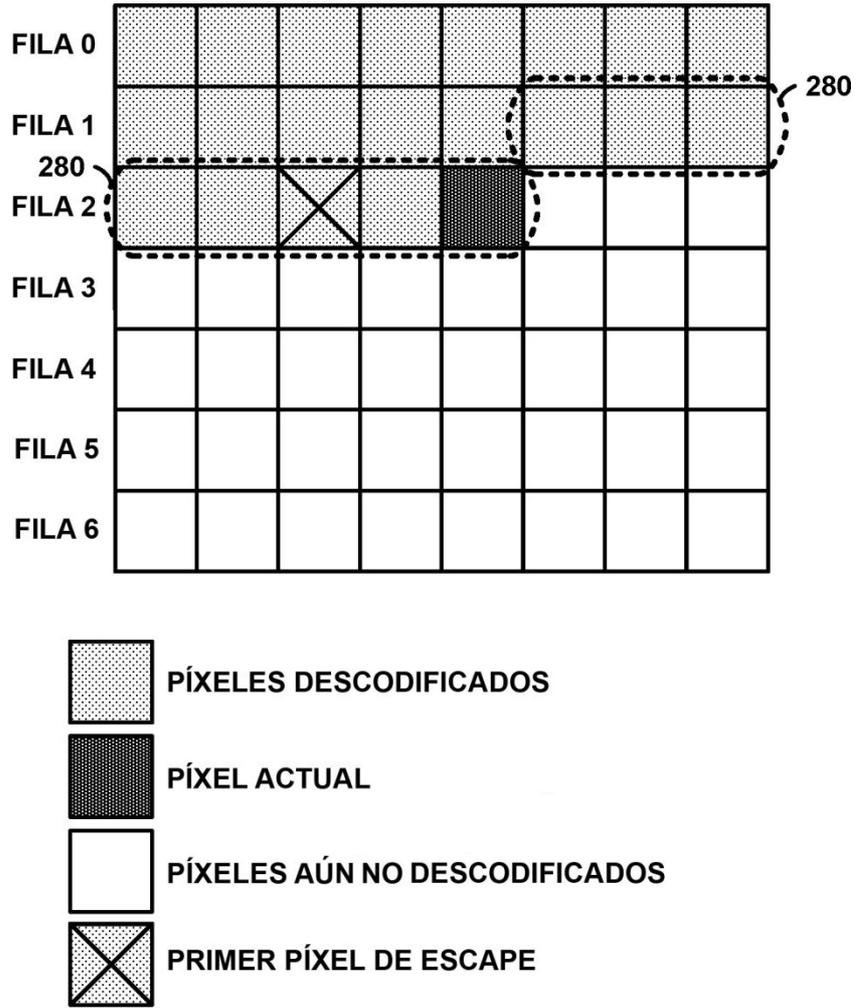


FIG. 6

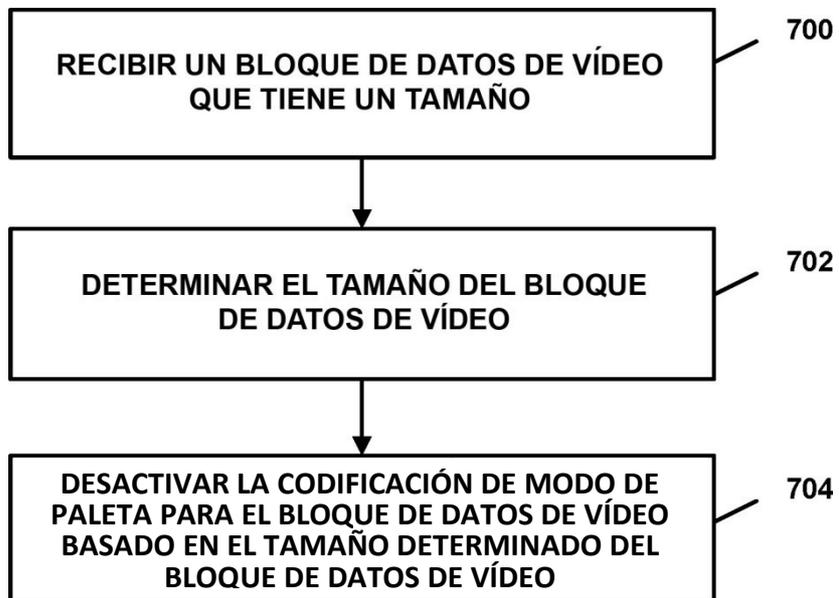


FIG. 7