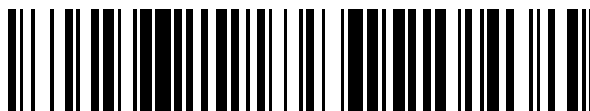


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 739 690**

51 Int. Cl.:

H04N 19/176 (2014.01)

H04N 19/196 (2014.01)

H04N 19/593 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **25.01.2016 PCT/US2016/014760**

87 Fecha y número de publicación internacional: **04.08.2016 WO16123033**

96 Fecha de presentación y número de la solicitud europea: **25.01.2016 E 16704738 (0)**

97 Fecha y número de publicación de la concesión europea: **08.05.2019 EP 3251356**

54 Título: **Codificación en modo paleta para codificación de vídeo**

30 Prioridad:

29.01.2015 US 201562109568 P
22.01.2016 US 201615004508

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
03.02.2020

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
Attn: International IP Administration, 5775
Morehouse Drive
San Diego, CA 92121-1714, US

72 Inventor/es:

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

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 739 690 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

Codificación en modo paleta para codificación de vídeo

5 **[0001]** Esta solicitud reivindica la prioridad de la solicitud provisional de Estados Unidos n.º 62/109.568 presentada el 29 de enero de 2015.

CAMPO TÉCNICO

10 **[0002]** Esta divulgación se refiere a la codificación y a la decodificación de vídeo.

ANTECEDENTES

15 **[0003]** Las capacidades del vídeo digital pueden incorporarse a una amplia gama de dispositivos, que incluye televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, 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», dispositivos de videoconferencia, dispositivos de transmisión de vídeo en tiempo real 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 eficiencia (HEVC) actualmente en desarrollo y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital con más eficacia implementando dichas técnicas de compresión de vídeo.

25 **[0004]** Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (entre imágenes) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (es decir, una trama de vídeo o una parte de una trama de vídeo) puede dividirse 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 adyacentes en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques adyacentes en la misma imagen o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.

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 que se va 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 una modalidad de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxel a un dominio de transformada, dando como resultado coeficientes residuales, que pueden cuantificarse entonces. Los coeficientes cuantificados, inicialmente dispuestos en una matriz bidimensional, pueden escanearse con el fin de producir un vector unidimensional de coeficientes, y puede aplicarse la codificación por entropía para lograr incluso más compresión.

45 **[0006]** El documento US2015/0016501 divulga técnicas para generar una paleta actual para un bloque actual, según el cual un decodificador de vídeo determina primero una o más entradas de paleta en una paleta predictiva que se copian a la paleta actual, y luego determina un número de nuevas entradas de paleta que no están en la paleta predictiva pero que están incluidas en la paleta actual.

SUMARIO

55 **[0007]** Esta divulgación se refiere a técnicas de codificación y decodificación de vídeo. En particular, esta divulgación describe técnicas para codificar y decodificar datos de vídeo con un modo de codificación basada en paleta. En un modo de codificación basada en paleta, los valores de píxel para un bloque de datos de vídeo pueden codificarse en relación con una paleta de valores de color asociados con el bloque de datos de vídeo. La paleta de valores de color puede ser determinada por un codificador de vídeo y puede contener los valores de color más comunes para un bloque en particular. El codificador de vídeo puede asignar un índice en la paleta de valores de color a cada píxel en el bloque de datos de vídeo, y señalar dicho índice a un decodificador de vídeo en un flujo de bits de vídeo codificado. El decodificador de vídeo puede luego usar el índice en la paleta para determinar qué valor de color usar para un píxel particular en el bloque.

65 **[0008]** Además de los índices de señalización en la paleta, un codificador de vídeo también puede transmitir la propia paleta en el flujo de bits de vídeo codificado. Las técnicas para transmitir la paleta pueden incluir la señalización explícita de los valores de la paleta, así como la predicción de las entradas de la paleta para un bloque

actual a partir de las entradas de la paleta de uno o más bloques codificados previamente. Esta divulgación describe técnicas para codificar paletas, incluyendo técnicas para codificar elementos de sintaxis relacionados con la codificación de paletas y/o la predicción de paletas.

5 **[0009]** En un ejemplo de la divulgación, un procedimiento de descodificación de datos de vídeo comprende recibir un bloque de datos de vídeo en un flujo de bits de vídeo codificado, habiendo sido codificado el bloque de datos de vídeo usando un modo de codificación basada en paleta, recibir una pluralidad de elementos de sintaxis que son indicativos de una paleta que se usó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado, en el que el primer elemento de sintaxis se codifica utilizando uno o más códigos de Golomb, de modo que la longitud del primer elemento de sintaxis codificada es menor o igual a un número máximo predeterminado de bits, descodificar la pluralidad de elementos de sintaxis, incluida la descodificación del primer elemento de sintaxis utilizando el uno o más códigos de Golomb, reconstruir la paleta basada en la pluralidad de elementos de sintaxis descodificados, y descodificar el bloque de datos de vídeo utilizando la paleta reconstruida.

10
15
20 **[0010]** En otro ejemplo de la divulgación, un procedimiento para codificar datos de vídeo comprende codificar un bloque de datos de vídeo usando un modo de codificación basada en paleta y una paleta, generar una pluralidad de elementos de sintaxis que son indicativos de la paleta que se usó para codificar el bloque de los datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado, codificar el primer elemento de sintaxis utilizando uno o más códigos Golomb de tal manera que la longitud del primer elemento de sintaxis codificado es menor o igual que un número máximo predeterminado de bits, e incluir la pluralidad de elementos de sintaxis en un flujo de bits de vídeo codificado.

25
30 **[0011]** En otro ejemplo de la divulgación, un aparato configurado para descodificar datos de vídeo comprende una memoria configurada para almacenar un flujo de bits de vídeo codificado, y un descodificador de vídeo configurado para recibir un bloque de datos de vídeo en el flujo de bits de vídeo codificado, habiendo sido codificado el bloque de datos de vídeo usando un modo de codificación basada en paleta, recibir una pluralidad de elementos de sintaxis que son indicativos de una paleta que se usó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señala explícitamente en el flujo de bits de vídeo codificado, en el que el primer elemento de sintaxis se codifica utilizando uno o más códigos de Golomb de tal manera que la longitud del primer elemento de sintaxis codificado es menor o igual a un número máximo predeterminado de bits, descodificar la pluralidad de elementos de sintaxis, incluida la descodificación del primer elemento de sintaxis utilizando uno o más códigos de Golomb, reconstruir la paleta basándose en la pluralidad de elementos de sintaxis descodificados, y descodificar el bloque de datos de vídeo usando la paleta reconstruida.

35
40 **[0012]** En otro ejemplo de la divulgación, un aparato configurado para codificar datos de vídeo comprende una memoria configurada para almacenar un bloque de datos de vídeo, y un codificador de vídeo configurado para codificar el bloque de datos de vídeo usando un modo de codificación basada en paleta y una paleta, generar una pluralidad de elementos de sintaxis que son indicativos de la paleta que se usó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado, codificar el primer elemento de sintaxis utilizando uno o más códigos de Golomb, de manera que la longitud del primer elemento de sintaxis codificado sea menor o igual a un número máximo predeterminado de bits, e incluir la pluralidad de elementos de sintaxis en un flujo de bits de vídeo codificado.

45
50 **[0013]** En otro ejemplo de la divulgación, un aparato configurado para descodificar datos de vídeo comprende medios para recibir un bloque de datos de vídeo en un flujo de bits de vídeo codificado, habiéndose codificado el bloque de datos de vídeo usando un modo de codificación basada en paleta, medios para recibir una pluralidad de elementos de sintaxis que son indicativos de una paleta que se usó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado, en el que el primer elemento de sintaxis se codifica utilizando uno o más códigos de Golomb, de modo que la longitud del primer elemento de sintaxis codificado sea menor o igual a un número máximo predeterminado de bits, medios para descodificar la pluralidad de elementos de sintaxis, incluida la descodificación del primer elemento de sintaxis que utiliza uno o más códigos de Golomb, medios para reconstruir la paleta basándose en la pluralidad de elementos de sintaxis descodificados, y medios para descodificar el bloque de datos de vídeo usando la paleta reconstruida.

55
60
65 **[0014]** En otro ejemplo de la divulgación, un aparato configurado para codificar datos de vídeo comprende medios para codificar un bloque de datos de vídeo usando un modo de codificación basada en paleta y una paleta, medios para generar una pluralidad de elementos de sintaxis que son indicativos de la paleta que fue utilizada para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado, medios para codificar el primer elemento de sintaxis usando uno o más códigos de Golomb de modo

que la longitud del primer elemento de sintaxis codificado sea menor o igual a un número máximo predeterminado de bits, y medios para incluir la pluralidad de elementos de sintaxis en un flujo de bits de vídeo codificado.

5 [0015] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que
 almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo configurado
 para descodificar datos de vídeo reciban un bloque de datos de vídeo en un flujo de bits de vídeo codificado,
 habiendo sido codificado el bloque de datos de vídeo utilizando un modo de codificación basada en paleta, reciban
 una pluralidad de elementos de sintaxis que son indicativos de una paleta que se usó para codificar el bloque de
 10 datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un
 número de valores de una paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo
 codificado, en el que el primer elemento de sintaxis se codifica utilizando uno o más códigos de Golomb de tal
 manera que la longitud del primer elemento de sintaxis codificado es menor o igual a un número máximo
 predeterminado de bits, descodifique la pluralidad de elementos de sintaxis, incluida la descodificación del primer
 15 elemento de sintaxis utilizando uno o más códigos de Golomb, reconstruya la paleta basándose en la
 descodificación de la pluralidad de elementos de sintaxis descodificados y descodifique el bloque de datos de vídeo
 usando la paleta reconstruida.

[0016] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que
 almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo configurado
 20 para codificar datos de vídeo codifiquen un bloque de datos de vídeo usando un modo de codificación basada en
 paleta y una paleta, generen una pluralidad de elementos de sintaxis que son indicativos de la paleta que se usó
 para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento
 de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de
 25 bits de vídeo codificado, codifiquen el primer elemento de sintaxis utilizando uno o más códigos Golomb de tal
 manera que la longitud del primer elemento de sintaxis codificado es menor o igual a un número máximo
 predeterminado de bits, e incluyan la pluralidad de elementos de sintaxis en un vídeo codificado flujo de bits.

[0017] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción.
 30 Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir
 de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

35 [0018]
 La figura 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo ejemplar que puede
 utilizar las técnicas descritas en esta divulgación.

40 La figura 2 es un diagrama de bloques que ilustra un codificador de vídeo ejemplar que puede implementar las
 técnicas descritas en esta divulgación.

La figura 3 es un diagrama de bloques que ilustra un descodificador de vídeo ejemplar que puede implementar
 las técnicas descritas en esta divulgación.

45 La figura 4 es un diagrama de bloques que ilustra una unidad de codificación basada en paleta ejemplar del
 codificador de vídeo de la figura 2.

50 La figura 5 es un diagrama conceptual que ilustra una técnica de predicción de paleta ejemplar de acuerdo con
 las técnicas de la divulgación.

La figura 6 es un diagrama conceptual que ilustra una técnica de codificación de vectores de predicción binarios
 de ejemplo de acuerdo con las técnicas de la divulgación.

55 La figura 7 es un diagrama de bloques que ilustra una unidad de descodificación basada en paleta ejemplar del
 codificador de vídeo de la figura 3.

La figura 8 es un diagrama de flujo que ilustra un procedimiento de codificación de vídeo ejemplar de acuerdo
 con las técnicas de esta divulgación.

60 La figura 9 es un diagrama de flujo que ilustra un procedimiento de descodificación de vídeo ejemplar de
 acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

65 [0019] Esta divulgación está relacionada con el campo de la codificación de vídeo, y más particularmente con la
 predicción o codificación de un bloque de datos de vídeo en modo de codificación basada en paleta. En la

codificación de vídeo tradicional, se supone que las imágenes son de tono continuo y espacialmente fluidas. Basándose en estas suposiciones, se han desarrollado diversas herramientas, tales como la transformada basada en bloques, el filtrado, etc., y dichas herramientas han demostrado un buen rendimiento para vídeos de contenido natural. Sin embargo, en aplicaciones como escritorio remoto, trabajo colaborativo y pantalla inalámbrica, el contenido de la pantalla generado por ordenador (por ejemplo, tales como texto o gráficos de ordenador) puede ser el contenido dominante que se vaya a comprimir. Este tipo de contenido tiende a tener tonos discretos, y presenta líneas nítidas y límites de objetos de alto contraste. Es posible que la suposición de tono continuo y homogeneidad ya no pueda aplicarse al contenido de la pantalla, y por tanto las técnicas tradicionales de codificación de vídeo pueden no ser eficientes de comprimir los datos de vídeo, incluido el contenido de la pantalla.

[0020] Esta divulgación describe la codificación basada en paleta, que puede ser particularmente adecuada para la codificación de contenido de pantalla. Por ejemplo, suponiendo que un área particular de datos de vídeo tiene un número relativamente pequeño de colores, un codificador de vídeo (por ejemplo, un codificador de vídeo o descodificador de vídeo) puede formar una llamada "paleta" para representar los datos de vídeo del área particular. La paleta puede expresarse como una tabla de colores o valores de píxeles que representan los datos de vídeo del área en particular (por ejemplo, un bloque dado). 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 uno o más valores de píxeles que se produzcan 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 paleta, el codificador de vídeo puede codificar valores de índice indicativos de uno o más de los valores de píxeles del bloque actual, en lugar de codificar valores de píxeles reales o sus residuos para un bloque actual de datos de vídeo. En el contexto de la codificación basada en paletas, los valores de índice indican entradas respectivas en la paleta que se usan para representar valores de píxeles individuales del bloque actual.

[0021] Por ejemplo, el codificador de vídeo puede codificar un bloque de datos de vídeo determinando la 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 uno o más de los valores de píxeles, y codificando el bloque con valores de índice 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 índice 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 índice para los píxeles individuales del bloque. El descodificador de vídeo puede relacionar los valores de índice de los píxeles con las entradas de la paleta para reconstruir los diversos valores de píxeles del bloque.

[0022] De acuerdo con los diversos ejemplos analizados a continuación, esta divulgación describe técnicas para mejorar la eficiencia de codificación al codificar bloques de datos de vídeo con un modo de codificación basada en paleta. Los ejemplos de esta divulgación incluyen técnicas para codificar datos de vídeo usando un modo de codificación basada en paleta, y técnicas para codificar elementos de sintaxis relacionados con el modo de codificación basada en paleta. En algunos ejemplos, las técnicas de esta divulgación se relacionan con la codificación de elementos de sintaxis utilizados por un descodificador de vídeo para determinar y/o reconstruir una paleta para un bloque de datos de vídeo.

[0023] En algunos ejemplos, las técnicas de codificación basadas en paleta de esta divulgación pueden configurarse para su uso con una o más normas de codificación de vídeo. Algunas normas de codificación de vídeo de ejemplo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de codificación de vídeo escalable (SVC) y de codificación de vídeo multivista (MVC). En otro ejemplo, las técnicas de codificación basadas en paleta pueden configurarse para su uso con la codificación de vídeo de alta eficiencia (HEVC). La HEVC es una nueva norma de codificación de vídeo desarrollada por el Equipo Colaborativo Conjunto en Codificación de vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) del ITU-T y el Grupo de Expertos en Imagen en Movimiento (MPEG) de ISO/IEC.

[0024] Recientemente, el Equipo Colaborativo Conjunto sobre Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC han terminado el diseño de la HEVC. La última especificación de HEVC, denominada en adelante HEVC Versión 1 o HEVC1, se describe en "ITU-T H.265 (V1)", que a partir del 24 de marzo de 2015 está disponible en <http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11885&lang=en>. El documento ITU-T H.265, SERIE H: SISTEMAS AUDIOVISUALES Y MULTIMEDIA, Infraestructura de Servicios Audiovisuales-Codificación de Vídeo en Movimiento, "High Efficiency Video Coding [Codificación de Vídeo de Alta Eficiencia]", abril de 2013, describe asimismo la norma HEVC. Una especificación reciente de extensiones de rango, denominada RExt en adelante, se describe en "ITU-T H.265 (V2)", que a partir del 24 de marzo de 2015 está disponible en <http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=12296&lang=en>.

[0025] Para proporcionar una codificación más eficiente del contenido generado por la pantalla, el JCT-VC está desarrollando una extensión de la norma HEVC, que se conoce como la norma de codificación de contenido de

pantalla (SCC) del HEVC. Un borrador de trabajo reciente de la norma SCC de HEVC, denominado "HEVC SCC Draft 2 [Borrador 2 de SCC de HEVC]" o "WD2", se describe en el documento JCTVC-S1005, R. Joshi y J. Xu, "HEVC screen content coding draft text 2 [Texto del borrador 2 de codificación de contenido de pantalla de HEVC]", Equipo de colaboración conjunto sobre codificación de vídeo (JCT-VC) del ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 19ª reunión: Estrasburgo, FR, 17-24 de octubre de 2014.

[0026] La figura 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 para codificación de vídeo basado en paleta. 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 a 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 configurarse para realizar técnicas de codificación de vídeo basada en paleta 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 selectivamente varios bloques de datos de vídeo, tales como las CU o PU en la codificación HEVC, usando la codificación basada en paleta o bien la codificación no basada en paleta. Las modalidades de codificación no basada en paleta pueden referirse a diversas modalidades de codificación temporal interpredictiva o a modalidades de codificación espacial intrapredictiva, tales como las diversas modalidades de codificación especificadas por la norma HEVC. Sin embargo, debe entenderse que las técnicas de esta divulgación pueden usarse con cualquier técnica de codificación de vídeo y/o normas que usan un modo de codificación basada en paleta.

[0027] Como se muestra en la figura 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 puede denominarse dispositivo de codificación de vídeo o aparato de codificación de vídeo. El dispositivo de destino 14 puede descodificar datos de vídeo codificados, generados por el dispositivo de origen 12. Por consiguiente, el dispositivo de destino 14 puede denominarse 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.

[0028] El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, dispositivos informáticos móviles, ordenadores plegables (es decir, portátiles), ordenadores de tableta, descodificadores, equipos telefónicos manuales tales como los denominados teléfonos "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores de coche o similares.

[0029] El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 a través de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de mover los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicación que permitan 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 inalámbricos y/o cableados, tales como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. 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 extensa o una red global (por ejemplo, Internet). Los uno o más medios de comunicación pueden incluir encaminadores, conmutadores, estaciones base u otros equipos que faciliten la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

[0030] En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacene 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 a través del acceso al disco o del acceso a la tarjeta. El medio de almacenamiento puede incluir varios medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria flash u otros medios adecuados de almacenamiento digital para almacenar datos de vídeo codificados.

[0031] En un ejemplo adicional, el canal 16 puede incluir un servidor de ficheros u otro dispositivo de almacenamiento intermedio que almacene 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 ficheros o en otro dispositivo de almacenamiento intermedio mediante transmisión continua 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. Entre los ejemplos de servidores de archivos se incluyen servidores de red (por ejemplo, para una página web), servidores del protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados a red (NAS) y unidades de disco local.

[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. Entre los ejemplos de tipos de conexiones de datos pueden incluirse canales inalámbricos (por ejemplo, conexiones WiFi), conexiones cableadas (por ejemplo, DSL, módem de cable, etc.) o combinaciones de ambos que sean adecuadas para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0033] Las técnicas de esta divulgación para la codificación de vídeo basada en paleta no están limitadas a aplicaciones o a configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo como soporte de varias aplicaciones multimedia, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, mediante 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 configurarse para soportar la transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

[0034] El sistema de codificación de vídeo 10 ilustrado en la figura 1 es simplemente un ejemplo. Las técnicas de esta divulgación pueden aplicarse a casos de uso de la codificación de vídeo (por ejemplo, codificación de vídeo o descodificación de vídeo) que no incluyen necesariamente alguna comunicación de datos entre los dispositivos de codificación y descodificación. En otros ejemplos, los datos se recuperan desde una memoria local, se transmiten continuamente por una red o similar. Un dispositivo de codificación de vídeo puede codificar y almacenar datos en la memoria, y/o un dispositivo de descodificación de vídeo puede recuperar y descodificar datos desde la memoria. En muchos ejemplos, la codificación y la descodificación se realiza mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en la memoria y/o recuperan y descodifican datos desde la memoria.

[0035] En el ejemplo de la figura 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.

[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 mediante 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 un acceso posterior mediante el dispositivo de destino 14 para su descodificación y/o su reproducción.

[0037] En el ejemplo de la figura 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 los datos de vídeo codificados por el 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 los datos de vídeo descodificados. El dispositivo de visualización 32 puede comprender varios 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.

[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 elementos de sintaxis en un medio de almacenamiento legible por ordenador en un flujo de bits codificado en el momento de la codificación, que entonces pueden ser recuperado por un dispositivo de descodificación en cualquier momento tras haber sido almacenado en este medio. Por lo tanto, aunque el descodificador de vídeo 30 puede mencionarse como "recibiendo" cierta información, la recepción de información no necesariamente se produce en tiempo real o casi real, y puede recuperarse desde un medio en algún momento después del almacenamiento.

[0039] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden, cada uno, implementarse como cualquiera entre varios circuitos adecuados, 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

parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado de almacenamiento no transitorio legible por ordenador, y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cualquiera de los anteriores (incluyendo hardware, software, una combinación de hardware y software, etc.) puede considerarse como uno o más procesadores. Cada uno entre el codificador de vídeo 20 y el descodificador de vídeo 30 se puede incluir en uno o más codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

[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. Además de la norma base de HEVC, hay esfuerzos constantes para producir extensiones de codificación de vídeo ajustable a escala, codificación de vídeo multivistas y codificación en 3D para HEVC. Además, pueden proporcionarse modalidades de codificación basada en paleta, por ejemplo, como se describe en esta divulgación, para la extensión de la norma HEVC. En algunos ejemplos, las técnicas descritas en esta divulgación para la 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 una modalidad de codificación basada en paletas para la codificación de unidades de codificación (CU) o unidades de predicción (PU) en un códec de HEVC se describe para los propósitos de ejemplo.

[0041] En la HEVC y en 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 como 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 solo incluir una matriz de muestras de luma.

[0042] Para generar una representación codificada de una imagen en HEVC, 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 correspondientes bloques de árbol de codificación de muestras de croma y estructuras sintácticas usadas para codificar las muestras de los bloques de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque de muestras de tamaño $N \times N$. Una CTU también puede denominarse "bloque de árbol" o "unidad de codificación de máximo tamaño" (LCU). Las CTU de HEVC pueden ser análogas en gran medida a los macrobloques de otras normas, tales como H.264/AVC. Sin embargo, una CTU no está limitada necesariamente a un tamaño 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 sintáctica 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 sintácticas usadas para codificar muestras del 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 algunos contextos, el análisis de las PU puede intercambiarse con el análisis de macrobloques o de particiones de macrobloques.

[0044] 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 de tamaño $N \times N$. Una CU puede ser un bloque de codificación de muestras de luma y dos correspondientes bloques de codificación de muestras de croma de una imagen que tenga una matriz de muestras de luma, una matriz de muestras de Cb y una matriz de muestras de Cr, y estructuras sintácticas 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 aplique la misma predicción. Una unidad de predicción (PU) de una CU puede ser un bloque de predicción de muestras de luma, dos correspondientes bloques de predicción de muestras de croma de una imagen y estructuras sintácticas usadas para predecir las muestras 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.

[0045] 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 la 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.

[0046] Si el codificador de vídeo 20 usa la 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 la 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.

[0047] 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 predictivos 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.

[0048] 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 transformación de luma, Cb y Cr). Un bloque de transformación puede ser un bloque rectangular de muestras en el que se aplique la misma transformación. Una unidad de transformación (TU) de una CU puede ser un bloque de transformación de muestras de luma, dos bloques de transformación correspondientes de muestras de croma y estructuras sintácticas usadas para transformar las muestras de bloques de transformación. De este modo, cada TU de una CU puede estar asociada a un bloque de transformación de luma, un bloque de transformación de Cb y un bloque de transformación de Cr. El bloque de transformación de luma asociado con la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformación de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformación de Cr puede ser un subbloque del bloque residual de Cr de la CU.

[0049] El codificador de vídeo 20 puede aplicar una o más transformaciones a un bloque de transformación para generar un bloque de coeficientes para una TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformación. Un coeficiente de transformación puede ser una cantidad escalar. Por ejemplo, el codificador de vídeo 20 puede aplicar una o más transformaciones a un bloque de transformación 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 transformaciones a un bloque de transformación 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 transformaciones a un bloque de transformación de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

[0050] 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 transformación se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformación, proporcionando una compresión adicional. 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 transformación cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar la Codificación Aritmética Binaria Adaptativa al Contexto (CABAC) en los elementos de sintaxis que indican los coeficientes de transformación cuantificados. El codificador de vídeo 20 puede emitir los elementos de sintaxis codificados por entropía en un flujo de bits. El flujo de bits también puede incluir elementos de sintaxis que no estén codificados por entropía.

[0051] El codificador de vídeo 20 puede emitir un flujo de bits que incluya los elementos de sintaxis codificados por entropía. El flujo de bits puede incluir una secuencia de bits que forme 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 de NAL incluye una cabecera de unidad de NAL y encapsula una carga útil de secuencia de octetos sin procesar (RBSP). La cabecera de la unidad de NAL puede incluir un elemento de sintaxis que indique un código de tipo de unidad de NAL. El código de tipo de unidad de NAL especificado por la cabecera de la unidad de NAL de una unidad de NAL indica el tipo de la unidad de NAL. Una RBSP puede ser una estructura sintáctica que contenga un número entero de octetos que se encapsule dentro de una unidad de NAL. En algunos casos, una RBSP incluye cero bits.

- 5 **[0052]** Diferentes tipos de unidades de 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 de NAL que encapsulan las RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes de SEI) pueden denominarse unidades de NAL de la capa de codificación de vídeo (VCL).
- 10 **[0053]** 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 obtener elementos de sintaxis del flujo de bits. Por ejemplo, el descodificador de vídeo 30 puede analizar sintácticamente el flujo de bits para descodificar elementos de sintaxis a partir 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 obtenidos (por ejemplo, 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 de muestras (es decir, bloques interpredictivos) para las PU de una CU actual. Además, el descodificador de vídeo 30 puede cuantificar inversamente bloques de coeficientes de transformación, asociados a las TU de la CU actual. El descodificador de vídeo 30 puede realizar transformaciones inversas en los bloques de coeficientes de transformación para reconstruir los bloques de transformación 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 de muestras para las PU de la CU actual a las muestras correspondientes de los bloques de transformación de las TU de la CU actual. Mediante la reconstrucción de los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.
- 15
- 20
- 25 **[0054]** En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse 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 representen los datos de vídeo de un área particular (por ejemplo, un bloque dado). De esta manera, en lugar de codificar los valores de píxeles reales o sus residuos para un bloque actual de datos de vídeo, el codificador de vídeo puede codificar valores de índice para uno o más de los valores de píxeles del bloque actual, donde los valores de índice indican entradas en la paleta que se usan para representar los valores de píxeles del bloque actual (por ejemplo, el índice puede correlacionarse con un conjunto de valores de Y, Cr y Cb o un conjunto de valores de R, G y B).
- 30
- 35 **[0055]** Por ejemplo, el codificador de vídeo 20 puede codificar un bloque de datos de vídeo determinando una paleta para el bloque, ubicando una entrada en la paleta que tenga un valor representativo del valor de uno o más píxeles individuales del bloque, y codificando el bloque con valores de índice que indiquen la entrada en la paleta usada para representar uno o más valores de píxeles individuales del bloque. Además, el codificador de vídeo 20 puede señalar los valores de índice en un flujo de bits codificado. A su vez, un dispositivo de descodificación de vídeo (por ejemplo, un descodificador de vídeo 30) puede obtener, a partir del flujo de bits codificado, la paleta para un bloque, así como los valores de índice usados para determinar los distintos píxeles individuales del bloque que usa la paleta. El descodificador de vídeo 30 puede relacionar los valores de índice de los píxeles individuales con las entradas de paleta para reconstruir los valores de píxeles del bloque. En los casos en que el valor de píxel de un píxel individual no es lo suficientemente cercano a ninguno de los valores de píxel representados por la paleta correspondiente para el bloque, el descodificador de vídeo 30 puede identificar dicho píxel individual como un píxel de escape, para los propósitos de la codificación basada en paletas. El valor de píxel de un píxel de escape puede codificarse explícitamente en lugar de a través de un índice de paleta.
- 40
- 45
- 50 **[0056]** 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 de índice) en la paleta que tenga un valor representativo del valor de uno o más de los valores residual de predicción de los píxeles individuales. Adicionalmente, el codificador de vídeo 20 puede codificar el bloque con valores de índice 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 índice para los valores residuales de predicción correspondientes a los píxeles individuales del bloque. Como se describe, los valores de índice 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 índice 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.
- 55
- 60
- 65 **[0057]** 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 varios números 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 escaneo. Para cada píxel incluido en el bloque dado, el codificador de vídeo 20 puede señalar el valor de índice que correlaciona el valor de píxel con una entrada correspondiente en la paleta. Si el valor del píxel no es lo suficientemente cercano al valor de ninguna de las entradas de la paleta (por ejemplo, un valor lo suficientemente cercano en comparación con algún umbral predeterminado), entonces dicho píxel se define como un "píxel de escape". De acuerdo con la codificación basada en paleta, el codificador de vídeo 20 puede codificar y señalar un valor de índice que está reservado para un píxel de escape, es decir, para indicar que es un píxel de escape y no un píxel para el cual hay una entrada en la paleta. En algunos ejemplos, el codificador de vídeo 20 también puede codificar y señalar el valor de píxel o un valor residual (o versiones cuantificadas del mismo) para un píxel de escape incluido en el bloque dado.

[0058] 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 índice 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 paleta es un píxel de escape, tal como determinando que el píxel está codificado por paleta con un valor de 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 paleta, el descodificador de vídeo 30 puede recibir el valor de 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 paleta correlacionando los valores de píxeles individuales con las entradas correspondientes de la paleta y usando el valor de píxel o el valor residual (o sus versiones cuantificadas) para reconstruir cualquier píxel de escape incluido en el bloque codificado por paleta.

[0059] El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden configurarse para operar de acuerdo con las técnicas descritas en esta divulgación, como se describirá con más detalle a continuación. En general, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden configurarse para codificar y descodificar datos de vídeo usando uno o más modos de codificación de paleta, en los que los modos de codificación de paleta no incluyen un modo de compartición de paleta. Las técnicas de esta divulgación incluyen un dispositivo de codificación de vídeo, tal como el codificador de vídeo 20, que se configura para determinar un primer bin de un primer elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente. El codificador de vídeo 20 puede configurarse además para codificar un flujo de bits. El flujo de bits puede incluir el primer elemento de sintaxis. El flujo de bits también puede no incluir un segundo elemento de sintaxis que indica un modo de compartición de paleta. En algunos ejemplos, determinar el primer bin del primer elemento de sintaxis comprende determinar el primer bin del primer elemento de sintaxis utilizando una codificación aritmética binaria adaptada al contexto. En otros ejemplos, la determinación del primer bin del primer elemento de sintaxis comprende la determinación del primer bin del primer elemento de sintaxis utilizando uno o más contextos. En algunos ejemplos de uso de uno o más contextos, el uno o más contextos pueden basarse en al menos uno de un número predicho de entradas de codificación de la paleta o un tamaño de bloque.

[0060] Además, esta divulgación describe que el codificador de vídeo 20 está configurado para determinar que un píxel actual es un primer píxel en una línea en un orden de escaneo. El codificador de vídeo 20 puede determinar además que un píxel adyacente situado sobre el píxel actual está disponible. En respuesta a la determinación de que el píxel actual es el primer píxel en la línea en el orden de escaneo y a la determinación de que el píxel adyacente situado sobre el píxel actual está disponible, el codificador de vídeo 20 puede configurarse además para omitir la codificación de un primer elemento de sintaxis en un flujo de bits, en el que el primer elemento de sintaxis indica un tipo de ejecución y codifica un resto del flujo de bits.

[0061] Además, las técnicas de esta divulgación incluyen el codificador de vídeo 20 que se configura para determinar un primer elemento de sintaxis que indica un tamaño máximo de paleta permitido y tiene un valor mínimo de cero. El codificador de vídeo 20 también se puede configurar para codificar un flujo de bits que incluya el primer elemento de sintaxis. En algunos ejemplos, el flujo de bits incluye además un segundo elemento de sintaxis que indica un tamaño máximo de la paleta de predicción y tiene un valor mínimo de cero. En algunos ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4096 y el segundo elemento de sintaxis tiene un valor máximo de 8192. En otros ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4095 y el segundo elemento de sintaxis tiene un valor máximo de 8191. En aún otros ejemplos, el primer elemento de sintaxis tiene un valor máximo que es igual a un número de píxeles en una unidad de codificación de máximo tamaño y el segundo elemento de sintaxis tiene un valor máximo que es igual a una constante positiva, como 2, multiplicada por el valor máximo del primer elemento de sintaxis. En otros ejemplos, el flujo de bits incluye otro elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente. En algunos ejemplos de esto, este elemento de sintaxis está representado por uno de un código Golomb Rice, un código exponencial Golomb, un código Rice truncado o un código unario. En otros

ejemplos de esto, este elemento de sintaxis se representa mediante uno de un código de Golomb Rice truncado, un código de exponencial Golomb truncado, un código truncado de Rice truncado, un código unario truncado o un código que también se usa para codificar un tercer elemento de sintaxis incluido en el flujo de bits codificado que indica si un índice de paleta se copia de un índice de paleta en una fila sobre un píxel actual o se codifica explícitamente en el flujo de bits codificado. En algunos ejemplos, este elemento de sintaxis está representado por un modo de Rice truncado. En algunos ejemplos, el elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente tiene un valor máximo que es igual al número de píxeles en un bloque actual de los datos de vídeo.

[0062] Además, esta divulgación describe un dispositivo de codificación de vídeo, tal como el descodificador de vídeo 30, que está configurado para recibir un flujo de bits codificado. El flujo de bits codificado no incluye un primer elemento de sintaxis que indica un modo de compartición de paleta. Además, el flujo de bits codificado incluye un segundo elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente. El descodificador de vídeo 30 puede configurarse además para descodificar un primer bin del segundo elemento de sintaxis. En algunos ejemplos, la descodificación del primer bin del segundo elemento de sintaxis comprende la descodificación del primer bin del segundo elemento de sintaxis utilizando un elemento de codificación aritmética binaria adaptable al contexto. En otros ejemplos, la descodificación del primer bin del segundo elemento de sintaxis comprende la descodificación del primer bin del segundo elemento de sintaxis utilizando uno o más contextos. En algunos ejemplos de uso de uno o más contextos, el uno o más contextos pueden basarse en al menos uno de un número predicho de entradas de codificación de la paleta o un tamaño de bloque.

[0063] Además, las técnicas de esta divulgación incluyen el descodificador de vídeo 30 que está configurado para recibir un flujo de bits codificado. El flujo de bits codificado puede incluir un primer elemento de sintaxis que indica un tipo de ejecución. El descodificador de vídeo 30 también puede configurarse para determinar que un píxel actual es un primer píxel en una línea en un orden de escaneo. El descodificador de vídeo 30 puede determinar además que un píxel adyacente situado sobre el píxel actual está disponible. En respuesta a la determinación de que un píxel actual es un primer píxel en una línea en un orden de escaneo y a la determinación de que un píxel adyacente situado sobre el píxel actual está disponible, el descodificador de vídeo 30 puede omitir la descodificación del primer elemento de sintaxis.

[0064] Además, las técnicas de esta divulgación incluyen el descodificador de vídeo 30 que está configurado para recibir un flujo de bits codificado que incluye un primer elemento de sintaxis que indica un tamaño máximo permitido de la paleta y tiene un valor mínimo de cero. El descodificador de vídeo 30 puede configurarse además para descodificar el flujo de bits codificado. En algunos ejemplos, el flujo de bits codificado incluye además un segundo elemento de sintaxis que indica un tamaño máximo de paleta de predicción y tiene un valor mínimo de cero. En algunos ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4096 y el segundo elemento de sintaxis tiene un valor máximo de 8192. En otros ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4095 y el segundo elemento de sintaxis tiene un valor máximo de 4095. En otros ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4095 y el segundo elemento de sintaxis tiene un valor máximo de 8191. En aún otros ejemplos, el primer elemento de sintaxis tiene un valor máximo que es igual a un número de píxeles en una unidad de codificación de máximo tamaño y el segundo elemento de sintaxis tiene un valor máximo que es igual a una constante positiva, como 2, multiplicada por el valor máximo del primer elemento de sintaxis. En otros ejemplos, el flujo de bits codificado incluye otro elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente. En algunos ejemplos, el elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente se representa mediante uno de un código Golomb Rice, un código Exponencial Golomb, un código Rice truncado o un código unario. En otros ejemplos, el elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente se representa mediante uno de un código de Golomb Rice truncado, un código de Exponencial Golomb truncado, un código truncado de Rice truncado, un código unario truncado el mismo código que se usa para codificar un elemento de sintaxis que indica si un índice de paleta se copia de un índice de paleta en una fila sobre un píxel actual o se codifica explícitamente en el flujo de bits codificado. En algunos ejemplos, el elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente se representa mediante un modo de Rice truncado. En algunos ejemplos, el elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente tiene un valor máximo que es igual al número de píxeles en un bloque actual de los datos de vídeo.

[0065] En otro ejemplo de la divulgación, el descodificador de vídeo 30 puede configurarse para recibir un bloque de datos de vídeo en un flujo de bits de vídeo codificado, habiendo sido codificado el bloque de datos de vídeo usando un modo de codificación basada en paleta, recibir una pluralidad de elementos de sintaxis que son indicativos de una paleta que se usó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado, descodificar la pluralidad de los elementos de sintaxis, incluida la descodificación del primer elemento de sintaxis utilizando uno o más códigos de Golomb, reconstruir la paleta basándose en la pluralidad de elementos de sintaxis descodificada, y descodificar el bloque de datos de vídeo utilizando la paleta reconstruida.

5 **[0066]** En otro ejemplo de la divulgación, el codificador de vídeo 20 puede configurarse para codificar un bloque de datos de vídeo usando un modo de codificación basada en paleta y una paleta, generar una pluralidad de elementos de sintaxis que son indicativos de la paleta que se usó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado, codificar el primer elemento de sintaxis utilizando uno o más códigos de Golomb, e incluir la pluralidad de elementos de sintaxis en un flujo de bits de vídeo codificado.

10 **[0067]** La figura 2 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que puede implementar diversas técnicas de esta divulgación. La figura 2 se proporciona con fines de explicación y no debería considerarse limitadora de las técnicas tales como las ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta divulgación describe el 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 **[0068]** En el ejemplo de la figura 2, el codificador de vídeo 20 incluye una memoria de datos de vídeo 98, una unidad de procesamiento de predicción 100, una unidad de generación residual 102, una unidad de procesamiento de transformación 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de procesamiento de transformación 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 procesamiento de predicción 100 incluye una unidad de procesamiento de interpredicción 120 y una unidad de procesamiento de intrapredicción 126. La unidad de procesamiento de interpredicción 120 incluye una unidad de estimación de movimiento y una unidad de compensación de movimiento (no mostrada). 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 estructurales.

20 **[0069]** La memoria de datos de vídeo 98 puede almacenar datos de vídeo a codificar por parte de los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 98 se pueden obtener, por ejemplo, a partir de la fuente de vídeo 18 de la figura 1. 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 modalidades de intracodificación o de intercodificación. La memoria de datos de vídeo 98 y la memoria intermedia de imágenes descodificadas 116 pueden estar formadas por cualquiera entre varios 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. La memoria de datos de vídeo 98 y la memoria intermedia de imágenes descodificadas 116 pueden proporcionarse mediante el mismo dispositivo de memoria o dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 98 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera de chip con respecto a esos componentes.

30 **[0070]** 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 asociarse a bloques de árbol de codificación (CTB) de luma de igual tamaño y a los CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede realizar la 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 procesamiento de predicción 100 puede dividir un CTB asociado a una CTU en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, etc.

35 **[0071]** 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 procesamiento de predicción 100 puede dividir los bloques de codificación asociados a la CU entre una o más PU de la CU. Por lo tanto, cada PU puede asociarse a un bloque de predicción de luma y a los correspondientes bloques de predicción de croma. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden dar soporte a las PU de diversos tamaños. Como se ha indicado anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU, y el tamaño de una PU puede referirse al tamaño de un bloque predictivo de luma de la PU. Suponiendo que el tamaño de una CU particular sea de $2N \times 2N$, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden dar soporte a tamaños de PU de $2N \times 2N$ o $N \times N$ para la intrapredicción, y a tamaños de PU simétricas de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares para la interpredicción. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden dar soporte a la partición asimétrica para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$, para la interpredicción.

40 **[0072]** 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 uno o más

bloques predictivos de muestras de la PU y la información de movimiento para la PU. La unidad de procesamiento de interpredicción 120 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento I, todas las PU son intrapredichas. Por lo tanto, si la PU está en un fragmento I, la unidad de procesamiento de interpredicción 120 no realiza interpredicción en la PU. Por tanto, para bloques codificados en el modo I, el bloque predictivo se forma usando predicción espacial a partir de bloques adyacentes previamente codificados dentro de la misma trama.

[0073] Si una PU está en un fragmento P, la unidad de estimación de movimiento de la unidad de procesamiento de interpredicción 120 puede buscar las imágenes de referencia en una lista de imágenes de referencia (por ejemplo, "RefPicListO") para 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 contenga bloques de muestra que se correspondan más estrechamente con los bloques de muestra de la PU. La unidad de estimación de movimiento 122 puede generar un índice de referencia que indique una posición en la RefPicListO de la imagen de referencia que contenga la región de referencia para la PU. Además, la unidad de estimación de movimiento puede generar un MV que indique 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 proporcione un desplazamiento de las coordenadas en la imagen descodificada actual a las coordenadas en una imagen de referencia. La unidad de estimación de movimiento puede emitir el índice de referencia y el vector de movimiento (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 muestras de la PU basándose en muestras reales o interpoladas en la ubicación de referencia, indicada por el MV de la PU.

[0074] 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 las imágenes de referencia de RefPicListO o de una segunda lista de imágenes de referencia ("RefPicList1") para una región de referencia para la PU. La unidad de estimación de movimiento puede emitir, como la información de movimiento de la PU, un índice de referencia que indique una posición en RefPicListO o en RefPicList1 de la imagen de referencia que contiene la región de referencia, un MV que indique un desplazamiento espacial entre un bloque de muestras de la PU y una ubicación de referencia asociada a la región de referencia, y uno o más indicadores de la dirección de predicción que indiquen si la imagen de referencia está en RefPicListO o en RefPicList1. La unidad de compensación de movimiento de la unidad de procesamiento de interpredicción 120 puede generar los bloques predictivos de muestras de la PU basándose, al menos en parte, en muestras reales (es decir, precisión de enteros) o interpoladas (es decir, precisión fraccionaria) en la región de referencia indicada por el vector de movimiento de la PU.

[0075] Para realizar la interpredicción bidireccional para una PU, la unidad de estimación de movimiento puede buscar las imágenes de referencia en la RefPicListO para una región de referencia para la PU y también puede buscar las imágenes de referencia en RefPicList1 para otra región de referencia para la PU. La unidad de estimación de movimiento puede generar índices de imágenes de referencia que indiquen posiciones en RefPicListO y en RefPicList1 de las imágenes de referencia que contienen las regiones de referencia. Además, la unidad de estimación de movimiento puede generar MV que indiquen 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 muestras 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.

[0076] De acuerdo con diversos ejemplos de esta divulgación, el codificador de vídeo 20 puede configurarse para realizar una codificación basada en paletas. Con respecto al entorno de HEVC, como ejemplo, las técnicas de codificación basada en paletas pueden configurarse para usarse como un modo CU. En otros ejemplos, las técnicas de codificación basada en paletas pueden configurarse para usarse como un modo PU en el entorno de la HEVC. Por consiguiente, todos los procesos divulgados, descritos en el presente documento (a lo largo de esta divulgación) en el contexto de un modo CU, pueden, adicionalmente o de forma alternativa, aplicarse a un modo PU. Sin embargo, estos ejemplos basados en la HEVC no deberían considerarse 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 por desarrollar. En estos casos, la unidad para la codificación por paleta pueden ser bloques cuadrados, bloques rectangulares o incluso regiones de forma no rectangular.

[0077] La unidad de codificación basada en paletas 122, por ejemplo, puede realizar una codificación 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 describen diversas funciones como realizadas

por la unidad de codificación basada en paletas 122, algunas de, o todas, estas funciones pueden realizarse por otras unidades de procesamiento, o una combinación de diferentes unidades de procesamiento.

[0078] La unidad de codificación basada en paletas 122 puede estar configurada para generar cualquiera de los diversos elementos de sintaxis descritos en el presente documento. Por consiguiente, el codificador de vídeo 20 puede configurarse para codificar bloques de datos de vídeo usando modos de codificación basada en paletas como se describe en esta divulgación. El codificador de vídeo 20 puede codificar selectivamente un bloque de datos de vídeo usando un modo de codificación por paleta, o codificar un bloque de datos de vídeo usando un modo diferente, por ejemplo, un modo de codificación HEVC interpredictiva o intrapredictiva. El bloque de datos de vídeo puede ser, por ejemplo, una CU o PU generada de acuerdo con un proceso de codificación de HEVC. Un codificador de vídeo 20 puede codificar algunos bloques con modos de predicción temporal interpredictiva o codificación espacial intrapredictiva y descodificar otros bloques con el modo de codificación basada en paleta.

[0079] 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 de muestras para la PU y diversos elementos de sintaxis. La unidad de procesamiento de intrapredicción 126 puede realizar la intrapredicción en las PU en fragmentos I, fragmentos P y fragmentos B.

[0080] Para realizar la 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. Cuando se usan algunos modos de intrapredicción para generar un conjunto de datos predictivos para la PU, la unidad de procesamiento de intrapredicción 126 puede extender los valores de las muestras de los bloques de muestra de las PU adyacentes a través de los bloques predictivos de la PU en direcciones asociadas con los modos de intrapredicción. Las PU adyacentes 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, de arriba a abajo, para las PU, CU y CTU. La unidad de procesamiento de intrapredicción 126 puede usar cualquiera de un número de diferentes 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.

[0081] La unidad de procesamiento de predicción 100 puede seleccionar los datos predictivos para las PU de una CU a partir de 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 procesamiento de predicción 100 selecciona los datos predictivos para las PU de la CU basándose en las mediciones de velocidad/distorsión de los conjuntos de datos predictivos. Los bloques de muestras predictivas de los datos predictivos seleccionados pueden denominarse en el presente documento como los bloques predictivos de muestras seleccionados.

[0082] La unidad de generación residual 102 puede generar, basándose en los bloques de codificación (por ejemplo, los bloques de codificación de luma, Cb y Cr) de una CU y los bloques predictivos de muestras seleccionados (por ejemplo, los bloques predictivos de luma, Cb y Cr) de las PU de la CU, bloques residuales (por ejemplo, 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 modo 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 correspondiente bloque predictivo de muestras seleccionado de una PU de la CU.

[0083] La unidad de procesamiento de transformación 104 puede realizar la partición de árbol cuaternario para dividir los bloques residuales asociados a una CU en bloques de transformación asociados a las TU de la CU. Por tanto, en algunos ejemplos, una TU puede asociarse con un bloque de transformación de luma y dos bloques de transformación de croma. Los tamaños y las posiciones de los bloques de transformación de luma y croma de las TU de una CU pueden o no basarse en los tamaños y posiciones de los bloques de predicción de las PU de la CU. Una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados con cada una de las regiones. Las TU de una CU pueden corresponder a nodos hoja del RQT.

[0084] La unidad de procesamiento de transformación 104 puede generar bloques de coeficientes de transformación para cada TU de una CU aplicando una o más transformaciones a los bloques de transformación de la TU. La unidad de procesamiento de transformación 104 puede aplicar diversas transformaciones a un bloque de transformación asociado a una TU. Por ejemplo, la unidad de procesamiento de transformación 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar al bloque de transformación. En algunos ejemplos, la unidad de procesamiento de transformación 104 no aplica transformadas a un bloque de transformación. En dichos ejemplos, el bloque de transformación puede tratarse como un bloque de coeficientes de transformación.

[0085] La unidad de cuantificación 106 puede cuantificar los coeficientes de transformación 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 transformación. Por ejemplo, un coeficiente de transformación de n bits puede redondearse hacia abajo hasta un coeficiente de transformación de m bits durante la cuantificación, donde n es mayor que m .

5 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 del QP asociado a la CU. La cuantificación puede introducir pérdida de información, por lo que los coeficientes de transformación cuantificados pueden tener una precisión inferior a los originales.

10 **[0086]** La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformación inversa 110 pueden aplicar la cuantificación inversa y transformaciones inversas al 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 de muestras generados por la unidad de procesamiento de predicción 100 para producir un bloque de transformación reconstruido asociado con una TU. Mediante la reconstrucción de bloques de transformación para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

15 **[0087]** La unidad de filtro 114 puede realizar una o más operaciones de desbloqueo para reducir los artefactos de bloqueo en los bloques de codificación asociados con una CU. 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 contenga los bloques de codificación reconstruidos para realizar la interpredicción en las PU de otras imágenes. Además, la unidad de procesamiento de intrapredicción 126 puede usar bloques de codificación reconstruidos en la memoria intermedia de imágenes descodificadas 116 para realizar la intrapredicción en otras PU en la misma imagen que la CU.

25 **[0088]** 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 procesamiento de predicción 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 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 por 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 incluya datos codificados por entropía, generados por la unidad de codificación por entropía 118. Por ejemplo, el flujo de bits puede incluir datos que representen un RQT para una CU.

40 **[0089]** En algunos ejemplos, la codificación residual no se realiza con la codificación de paleta. Por consiguiente, el codificador de vídeo 20 no puede realizar la transformación o la cuantificación cuando se codifique usando un modo de codificación por paleta. Además, el codificador de vídeo 20 puede codificar por entropía los datos generados usando un modo de codificación por paleta independientemente de los datos residuales.

45 **[0090]** 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 describió 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.

50 **[0091]** De acuerdo con una o más técnicas de esta divulgación, la unidad de codificación basada en paleta 122 puede aplicar técnicas de esta descripción para realizar un valor de muestra a la conversión de índices para codificar datos de vídeo usando uno o más modos de codificación de paleta, en la que los modos de codificación de paleta no incluyen un modo de compartición de paleta. Las técnicas de esta divulgación incluyen la unidad de codificación basada en paleta 122 del codificador de vídeo 20, que está configurada para determinar un primer bin de un primer elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente. La unidad de codificación basada en paleta 122 del codificador de vídeo 20 puede configurarse además para codificar un flujo de bits. El flujo de bits puede incluir el primer elemento de sintaxis. El flujo de bits también puede no incluir un segundo elemento de sintaxis que indica un modo de compartición de paleta. En algunos ejemplos, determinar el primer bin del primer elemento de sintaxis comprende determinar el primer bin del primer elemento de sintaxis utilizando una codificación aritmética binaria adaptada al contexto. En otros ejemplos, la determinación del primer bin del primer elemento de sintaxis comprende la determinación del primer bin del primer elemento de sintaxis utilizando uno o más contextos. En algunos ejemplos de uso de uno o más contextos, el uno o más contextos pueden basarse en al menos uno de un número predicho de entradas de codificación de paleta o un tamaño de bloque.

65 **[0092]** Además, las técnicas de esta divulgación incluyen la unidad de codificación basada en paleta 122 del codificador de vídeo 20 que se configura para determinar que un píxel actual es un primer píxel en una línea en un

orden de escaneo. La unidad de codificación basada en la paleta 122 del codificador de vídeo 20 puede determinar además que un píxel adyacente situado sobre el píxel actual está disponible. En respuesta a la determinación de que el píxel actual es el primer píxel en la línea en el orden de escaneo y a la determinación de que el píxel adyacente situado sobre el píxel actual está disponible, la unidad de codificación basada en paleta 122 del codificador de vídeo 20 puede configurarse aún más para omitir la codificación de un primer elemento de sintaxis en un flujo de bits, en el que el primer elemento de sintaxis indica un tipo de ejecución y codifica un resto del flujo de bits.

[0093] Además, las técnicas de esta divulgación incluyen la unidad de codificación basada en paleta 122 del codificador de vídeo 20 que está configurada para determinar un primer elemento de sintaxis que indica un tamaño máximo permitido de la paleta y tiene un valor mínimo de cero. La unidad de codificación basada en paleta 122 del codificador de vídeo 20 también puede configurarse para codificar un flujo de bits que incluya el primer elemento de sintaxis. En algunos ejemplos, el flujo de bits incluye además un segundo elemento de sintaxis que indica un tamaño máximo de la paleta de predicción y tiene un valor mínimo de cero. En algunos ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4096 y el segundo elemento de sintaxis tiene un valor máximo de 8192. En otros ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4095 y el segundo elemento de sintaxis tiene un valor máximo de 4095. En otros ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4095 y el segundo elemento de sintaxis tiene un valor máximo de 8191. En aún otros ejemplos, el primer elemento de sintaxis tiene un valor máximo que es igual a un número de píxeles en una unidad de codificación de máximo tamaño y el segundo elemento de sintaxis tiene un valor máximo que es igual a una constante positiva, como 2, multiplicada por el valor máximo del primer elemento de sintaxis. En otros ejemplos, el flujo de bits incluye otro elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente. En algunos ejemplos de esto, este elemento de sintaxis está representado por uno de un código de Golomb Rice, un código exponencial Golomb, un código de Rice truncado o un código unario. En otros ejemplos de esto, este elemento de sintaxis se representa mediante uno de un código de Golomb Rice truncado, un código de exponencial Golomb truncado, un código truncado de Rice truncado, un código unario truncado o un código que también se usa para codificar un tercer elemento de sintaxis incluido en el flujo de bits codificado que indica si un índice de paleta se copia de un índice de paleta en una fila sobre un píxel actual o se codifica explícitamente en el flujo de bits codificado. En algunos ejemplos, este elemento de sintaxis está representado por un modo de Rice truncado. En algunos ejemplos, el elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente tiene un valor máximo que es igual al número de píxeles en un bloque actual de los datos de vídeo.

[0094] La figura 3 es un diagrama de bloques que ilustra un descodificador de vídeo 30 a modo de ejemplo que está configurado para implementar las técnicas de esta divulgación. El descodificador de vídeo 30 puede operar de manera recíproca a la del codificador de vídeo 20 descrito con referencia a la figura 2. La figura 3 se proporciona para propósitos de explicación y no se limita a las técnicas como las ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta divulgación describe el 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 en los que se utiliza la codificación en modo paleta.

[0095] En el ejemplo de la figura 3, el descodificador de vídeo 30 incluye una memoria de datos de vídeo 148, una unidad de descodificación por entropía 150, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de procesamiento de transformación inversa 156, una unidad de reconstrucción 158, una unidad de filtro 160 y una memoria intermedia de imágenes descodificadas 162. La unidad de procesamiento de predicción 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 paleta 165 configurada para realizar diversos aspectos de las técnicas de codificación basada en paleta descritas en esta divulgación. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes estructurales.

[0096] La memoria de datos de vídeo 148 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 148 pueden obtenerse, por ejemplo, a partir del canal 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 148 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 almacene datos de vídeo de referencia para su uso en la descodificación de datos de vídeo mediante el descodificador de vídeo 30, por ejemplo, en modos de intracodificación o de intercodificación. La memoria de datos de vídeo 148 y la memoria intermedia de imágenes descodificadas 162 pueden estar formadas por cualquiera de entre varios 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. La memoria de datos de vídeo 148 y la memoria intermedia de imágenes descodificadas 162 pueden proporcionarse mediante el mismo dispositivo de memoria o dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 148

puede estar en un chip con otros componentes del decodificador de vídeo 30, o fuera de chip con respecto a esos componentes.

5 **[0097]** La memoria de datos de vídeo 148, es decir, una CPB, puede recibir y almacenar datos de vídeo codificados (por ejemplo, unidades de NAL) de un flujo de bits. La unidad de decodificación por entropía 150 puede recibir datos de vídeo codificados (por ejemplo, unidades de NAL) desde la memoria de datos de vídeo 148 y puede analizar sintácticamente las unidades de NAL para descodificar elementos de sintaxis. La unidad de decodificación por entropía 150 puede descodificar por entropía elementos de sintaxis codificados por entropía en las unidades de NAL. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154,
10 la unidad de procesamiento de transformación 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 obtenidos (por ejemplo, extraídos) del flujo de bits.

15 **[0098]** Las unidades de NAL del flujo de bits pueden incluir unidades de NAL de fragmentos codificados. Como parte de la decodificación del flujo de bits, la unidad de decodificación por entropía 150 puede extraer y descodificar por entropía elementos de sintaxis de las unidades de 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 en la cabecera de fragmento pueden incluir un elemento de sintaxis que identifique un PPS asociado a una imagen que
20 contenga el fragmento.

25 **[0099]** Además de descodificar los elementos de sintaxis del flujo de bits, el decodificador 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 decodificador 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 decodificador de vídeo 30 puede reconstruir bloques residuales de la CU.

30 **[0100]** Como parte de realizar una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede cuantificar de forma inversa, es decir, descuantificar, los bloques de coeficientes asociados a la TU. La unidad de cuantificación inversa 154 puede utilizar un valor de QP asociado a la CU de la TU para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa, para aplicar por la unidad de cuantificación inversa 154. 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 transformación. La relación de compresión también puede depender del
35 procedimiento de codificación por entropía empleado.

40 **[0101]** Después de que la unidad de cuantificación inversa 154 cuantifica inversamente un bloque de coeficientes, la unidad de procesamiento de transformación inversa 156 puede aplicar una o más transformaciones inversas al bloque de coeficientes para generar un bloque residual asociado a la TU. Por ejemplo, la unidad de procesamiento de transformación inversa 156 puede aplicar una DCT inversa, una transformación entera inversa, una transformación de Karhunen-Loeve (KLT) inversa, una transformación de rotación inversa, una transformación direccional inversa u otra transformación inversa al bloque de coeficientes.

45 **[0102]** Si se codifica una PU usando la intrapredicción, la unidad de procesamiento de intrapredicción 166 puede realizar la 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 contiguas. 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.
50

55 **[0103]** La unidad de procesamiento de predicción 152 puede construir una primera lista de imágenes de referencia (RefPicList0) y una segunda lista de imágenes de referencia (RefPicList1) basándose en los elementos de sintaxis extraídos del flujo de bits. Además, si se codifica una PU usando la interpredicción, la unidad de decodificación 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 en uno o más bloques de referencia para la PU, bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para la PU.

60 **[0104]** La unidad de reconstrucción 158 puede usar los bloques de transformación (por ejemplo, los bloques de transformación de luma, Cb y Cr) asociados con las TU de una CU y los bloques predictivos (por ejemplo, los bloques de luma, Cb y Cr) de las PU de la CU, por ejemplo, cualquiera de los datos de intrapredicción o de los datos de interpredicción, según corresponda, para reconstruir los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) de la CU. Por ejemplo, la unidad de reconstrucción 158 puede agregar muestras de los bloques de transformación (por ejemplo, los bloques de transformación de luma, Cb y Cr) a las muestras
65

correspondientes de los bloques predictivos (por ejemplo, los bloques predictivos de luma, Cb y Cr) para reconstruir los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) de la CU.

5 **[0105]** La unidad de filtro 160 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo asociados con los bloques de codificación (por ejemplo, los bloques de codificación de luma, Cb y Cr) de la CU. El descodificador de vídeo 30 puede almacenar los bloques de codificación (por ejemplo, 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 la posterior compensación de movimiento, la intrapredicción y la presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la figura 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en los bloques (por ejemplo, los bloques de luma, Cb y Cr) en la memoria intermedia de imágenes descodificadas 162, operaciones de intrapredicción o de interpredicción en las PU de otras CU. De esta forma, el descodificador de vídeo 30 puede obtener, a partir del flujo de bits, los niveles de coeficientes de transformación de un bloque de coeficientes significativo, cuantificar inversamente los niveles de coeficientes de transformación, aplicar una transformación a los niveles de coeficientes de transformación para generar un bloque de transformación, generar, basándose al menos en parte en el bloque de transformación, un bloque de codificación, y emitir el bloque de codificación para su visualización.

20 **[0106]** De acuerdo con diversos ejemplos de esta divulgación, el descodificador de vídeo 30 puede configurarse 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 seleccione una modalidad de codificación basada en paletas, por ejemplo, para una CU o PU. Por ejemplo, la unidad de descodificación basada en paletas 165 puede configurarse para generar una paleta que tenga entradas que indiquen valores de píxeles. Además, en este ejemplo, la unidad de descodificación basada en paletas 165 puede recibir información que asocie al menos algunas posiciones de un bloque de datos de vídeo con entradas en la paleta. En este ejemplo, la unidad de descodificación basada en paletas 165 puede seleccionar valores de píxeles en la paleta basándose en la información. Adicionalmente, en este ejemplo, la unidad de descodificación basada en paletas 165 puede reconstruir los valores de píxeles del bloque basándose en los valores de píxeles seleccionados. Aunque se describen diversas funciones como realizadas por la unidad de descodificación basada en paletas 165, algunas de, o todas, estas funciones pueden realizarse por otras unidades de procesamiento, o por una combinación de diferentes unidades de procesamiento.

35 **[0107]** De acuerdo con una o más técnicas de esta divulgación, la unidad de descodificación basada en paletas 165 puede recibir información de modo de codificación por paleta, y realizar las operaciones anteriores cuando la información de modo de codificación por paleta indique que el modo de codificación por paleta se aplica al bloque. Cuando la información del modo de codificación por paleta indica que el modo de codificación por paleta no se aplica al bloque, o cuando la información de otro modo indica el uso de un modo diferente, la unidad de descodificación basada en paletas 165 descodifica el bloque de datos de vídeo usando un modo de codificación no basada en paletas, por ejemplo, tal como un modo de codificación HEVC interpredictivo o intrapredictivo, cuando la información del modo de codificación por paleta indica que el modo de codificación por paleta no se aplica al bloque. El bloque de datos de vídeo puede ser, por ejemplo, una CU o PU generada de acuerdo con un proceso de codificación de HEVC. Un descodificador de vídeo 30 puede codificar algunos bloques con modos de predicción temporal interpredictiva o codificación espacial intrapredictiva y descodificar otros bloques con el modo de codificación basada en paletas. 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.

50 **[0108]** De acuerdo con una o más de las técnicas de esta divulgación, el descodificador de vídeo 30, y específicamente la unidad de descodificación basada en paletas 165, puede realizar la descodificación de vídeo basada en paletas de bloques de vídeo codificados por paleta. Como se describió anteriormente, una paleta descodificada por el descodificador de vídeo 30 puede codificarse y señalizarse explícitamente por el codificador de vídeo 20, reconstruirse por el descodificador de vídeo 30 con respecto a un bloque codificado por paleta recibido, predecirse a partir de entradas de paleta anteriores, predecirse a partir de valores de píxeles anteriores, o una combinación de los mismos.

55 **[0109]** La unidad de descodificación basada en paleta 165 puede aplicar técnicas de esta divulgación para realizar un valor de muestra a la conversión de índices para descodificar los datos de vídeo utilizando uno o más modos de codificación de paleta, en la que los modos de codificación de paleta no incluyen un modo de compartición de paleta. Además, las técnicas de esta divulgación incluyen una unidad de descodificación basada en paleta 165 del descodificador de vídeo 30 que está configurada para recibir un flujo de bits codificado. En este ejemplo, el flujo de bits codificado no incluye un primer elemento de sintaxis que indica un modo de compartición de paleta. Además, el flujo de bits codificado incluye un segundo elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente. La unidad de descodificación basada en la paleta 165 del descodificador de vídeo 30 puede configurarse adicionalmente para descodificar un primer bin del segundo elemento de sintaxis. En algunos ejemplos, la descodificación del primer bin del segundo elemento de sintaxis comprende la descodificación del primer bin del segundo elemento de sintaxis utilizando una unidad de codificación

aritmética binaria adaptable al contexto (CABAC). En otros ejemplos, la descodificación del primer bin del segundo elemento de sintaxis comprende la descodificación del primer bin del segundo elemento de sintaxis utilizando uno o más contextos. En algunos ejemplos de uso de uno o más contextos, el uno o más contextos pueden basarse en al menos uno de un número predicho de entradas de codificación de la paleta o un tamaño de bloque.

5

[0110] Además, las técnicas de esta divulgación incluyen una unidad de descodificación basada en paleta 165 del descodificador de vídeo 30 que está configurada para recibir un flujo de bits codificado. El flujo de bits codificado puede incluir un primer elemento de sintaxis que indica un tipo de ejecución. La unidad de descodificación basada en la paleta 165 del descodificador de vídeo 30 también puede configurarse para determinar que un píxel actual es un primer píxel en una línea en un orden de escaneo. La unidad de descodificación basada en paleta 165 del descodificador de vídeo 30 puede determinar además que un píxel adyacente situado encima del píxel actual está disponible. En respuesta a la determinación de que un píxel actual es un primer píxel en una línea en un orden de escaneo y a la determinación de que un píxel adyacente situado sobre el píxel actual está disponible, la unidad de descodificación basada en paleta 165 del descodificador de vídeo 30 puede omitir la descodificación del primer elemento de sintaxis.

10

15

[0111] Además, las técnicas de esta divulgación incluyen una unidad de descodificación basada en paleta 165 del descodificador de vídeo 30 que está configurada para recibir un flujo de bits codificado que incluye un primer elemento de sintaxis que indica un tamaño máximo permitido de la paleta y tiene un valor mínimo de cero. La unidad de descodificación basada en paleta 165 del descodificador de vídeo 30 puede configurarse además para descodificar el flujo de bits codificado. En algunos ejemplos, el flujo de bits codificado incluye además un segundo elemento de sintaxis que indica un tamaño máximo de paleta de predicción y tiene un valor mínimo de cero. En algunos ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4096 y el segundo elemento de sintaxis tiene un valor máximo de 8192. En otros ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4095 y el segundo elemento de sintaxis tiene un valor máximo de 4095. En otros ejemplos, el primer elemento de sintaxis tiene un valor máximo de 4095 y el segundo elemento de sintaxis tiene un valor máximo de 8191. En aún otros ejemplos, el primer elemento de sintaxis tiene un valor máximo que es igual a un número de píxeles en una unidad de codificación de máximo tamaño y el segundo elemento de sintaxis tiene un valor máximo que es igual a una constante positiva, como 2, multiplicada por el valor máximo del primer elemento de sintaxis. En otros ejemplos, el flujo de bits codificado incluye otro elemento de sintaxis, por ejemplo, un tercer elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente. En algunos ejemplos de esta divulgación, el elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente se representa mediante uno de un código de Golomb Rice, un código de exponencial Golomb, un código de Rice Truncado o un código unario. En otros ejemplos de esta divulgación, el elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente se representa mediante uno de un código de Golomb Rice truncado, un código exponencial Golomb truncado, un código truncado de Rice truncado, un código unario truncado, o un código que también se usa para codificar un tercer elemento de sintaxis incluido en el flujo de bits codificado que indica si un índice de paleta se copia de un índice de paleta en una fila sobre un píxel actual o se codifica explícitamente en el flujo de bits codificado. En algunos ejemplos, el elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente se representa mediante un modo de Rice truncado. En algunos ejemplos, el elemento de sintaxis que indica un número de entradas en una paleta actual que se señalizan explícitamente tiene un valor máximo que es igual al número de píxeles en un bloque actual de los datos de vídeo.

20

25

30

35

40

45

[0112] Ha de reconocerse que, según el ejemplo, ciertos actos o sucesos de cualquiera de las técnicas descritas en el presente documento pueden realizarse en una secuencia distinta, pueden añadirse, combinarse u omitirse por completo (por ejemplo, no todos los actos o sucesos descritos son necesarios para la puesta en práctica de las técnicas). Además, en determinados ejemplos, los actos o sucesos se pueden llevar a cabo simultáneamente, por ejemplo, mediante procesamiento de múltiples hilos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente. Además, si bien determinados aspectos de la presente divulgación se describen como que se llevan a cabo por un único módulo o unidad para propósitos de claridad, se debe entender que las técnicas de la presente divulgación se pueden llevar a cabo por una combinación de unidades o módulos asociados con un codificador de vídeo.

50

55

[0113] Ciertos aspectos de esta divulgación se han descrito con respecto a la norma de HEVC en desarrollo, con fines de ilustración. 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 patentados, aún no desarrollados.

60

[0114] Las técnicas descritas anteriormente pueden realizarse mediante el codificador de vídeo 20 (figuras 1 y 2) y/o el descodificador de vídeo 30 (figuras 1 y 3), de los cuales ambos pueden denominarse en general codificadores de vídeo. Del mismo modo, la codificación de vídeo puede referirse a la codificación de vídeo o a la descodificación de vídeo, según corresponda.

65

[0115] En algunos ejemplos, las técnicas de codificación basada en paletas pueden configurarse para su uso en uno o más modos de codificación de la norma HEVC o de la norma SCC de HEVC. En otros ejemplos, las técnicas

de codificación basada en paletas se pueden usar de forma independiente o como parte de otros sistemas o normas existentes o futuros. En algunos ejemplos, las técnicas para la codificación basada en paletas de datos de vídeo se pueden usar con una o más técnicas de codificación adicionales, tales como técnicas para la codificación interpredictiva o la codificación intrapredictiva de datos de vídeo. Por ejemplo, como se describe en mayor detalle a continuación, un codificador o descodificador, o un codificador-descodificador combinado (códec), se puede configurar para realizar una codificación interpredictiva e intrapredictiva, así como una codificación basada en paletas.

[0116] Con respecto al entorno de la HEVC, como ejemplo, las técnicas de codificación basada en paletas se pueden configurar para usarse como un modo de unidad de codificación (CU). En otros ejemplos, las técnicas de codificación basada en paletas pueden configurarse para usarse como un modo de unidad de predicción (PU) en el entorno de la HEVC. Por consiguiente, todos los procesos divulgados a continuación, descritos en el contexto de un modo CU, pueden, adicional o de forma alternativa, aplicarse a una PU. Sin embargo, estos ejemplos basados en la HEVC no deberían considerarse 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 por desarrollar. En estos casos, la unidad para la codificación por paleta pueden ser bloques cuadrados, bloques rectangulares o incluso regiones de forma no rectangular.

[0117] La idea básica de la codificación basada en paletas es que, para cada CU, se obtiene una paleta que comprende (y puede consistir en) los valores de píxeles más dominantes en la CU actual. El tamaño y los elementos de la paleta se transmiten primero desde un codificador de vídeo a un descodificador de vídeo. El tamaño y/o los elementos de la paleta pueden codificarse directamente o codificarse de manera predictiva usando el tamaño y/o los elementos de la paleta en las CU adyacentes (por ejemplo, la CU codificada superior y/o izquierda). Después de eso, los valores de píxeles en la CU se codifican basándose en la paleta de acuerdo con un cierto orden de escaneo. Para cada ubicación de píxeles en la CU, un indicador, por ejemplo, un indicador de paleta, se transmite primero para indicar si el valor de píxeles está incluido en la paleta. En algunos ejemplos, este indicador se llama `copy_above_palette_indices_flag`. Para esos valores de píxel que correlacionan con una entrada en la paleta, el índice de paleta asociado con esa entrada se señala para la ubicación de píxel dada en la CU. Para aquellos valores de píxel que no existen en la paleta, se puede asignar un índice especial al píxel y el valor de píxel real (en algunos casos, un valor de píxel cuantificado) se transmite para la ubicación de píxel dada en la CU. Estos píxeles se conocen como "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.

[0118] En otros ejemplos, no se utiliza ningún indicador para indicar explícitamente si un píxel es un píxel de "escape". En su lugar, se puede usar un indicador u otro elemento de sintaxis para indicar un tipo de ejecución. El elemento de sintaxis que indica el tipo de ejecución puede indicar si los siguientes índices se copian desde la posición sobre el píxel actual o si hay una ejecución de valores de índice señalizados. Si el valor de índice obtenido de un píxel particular corresponde a un "índice de escape" (por ejemplo, un índice predeterminado en la paleta que indica el uso de un píxel de escape), entonces el descodificador de vídeo 30 puede determinar que dicho píxel es un píxel de escape.

[0119] Para mejorar la eficiencia de la codificación del contenido de la pantalla, se han propuesto varios procedimientos que amplían el modo de paleta. Por ejemplo, tales procedimientos se pueden encontrar en JCTVC-S0114 (Kim, J. y otros, "CE6-related: Enabling copy above mode prediction at the boundary of CU [Relacionado con CE6: Habilitación de predicción del modo de copia superior en el límite de la CU]", Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 19ª Reunión: Estrasburgo, FR, 17-24 de octubre de 2014); JCTVC-S0120 (Ye, J. y otros, "Non-CE6: Copy previous mode [No CE6: Modo de copia anterior]", Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 19ª reunión: Estrasburgo, FR, 17-24 de octubre de 2014); y JCTVC-S0151 (Wang, W. y otros, "Non-CE6: 2-D Index Map Coding of Palette Mode in HEVC SCC [No CE6: codificación por correlación de índices 2-D de modo paleta en SCC de HEVC]", Equipo colaborativo conjunto sobre codificación de vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 19ª reunión: Estrasburgo, FR, 17-24 de octubre de 2014).

[0120] El documento X. Guo y A. Saxena, "RCE4: Summary report of HEVC Range Extension Core Experiments 4 (RCE4) on palette coding for screen content [RCE4: Informe resumido de experimentos esenciales de extensión de rango de HEVC 4 (RCE4)]", JCTVC-P0035, San José, EE. UU., 9-17 de enero de 2014 describe dos resultados de prueba de los modos basados en paleta, que se informó que lograron resultados significativos de reducción de la tasa de distorsión Bjontegaard (BD-rate), especialmente para los contenidos de la pantalla. Los dos procedimientos se resumen brevemente a continuación.

[0121] En un procedimiento de ejemplo, como se describe, por ejemplo, en el documento X. Guo, Y. Lu y S. Li, "RCE4: Test 1. Test 1. Major-color-based screen content coding [RCE4: Prueba 1. Codificación de contenido de pantalla basada en el color principal]", JCTVC-P0108, San José, EE. UU., 9-17 de enero de 2014, se utiliza un

algoritmo basado en histograma para clasificar los píxeles. En particular, los valores de pico N más significativos en un histograma se seleccionan como colores principales para la codificación. Los valores de píxel que están cerca de un color principal se cuantificarán al color principal. Otros píxeles que no pertenecen a ningún conjunto de colores principales son los píxeles de escape, que también se cuantifican antes de la codificación. Para la codificación sin pérdida, no se utiliza cuantificación.

[0122] Al utilizar la clasificación, los píxeles de una unidad de codificación (CU) se pueden convertir en índices de color. Después de eso, el número de color principal y los valores se codifican. Luego, los índices de color se codifican de la siguiente manera:

- Para cada línea de píxel, se señala un indicador para indicar el modo de codificación. Hay tres modos: modo horizontal, modo vertical y modo normal.

- Si el modo es horizontal, la línea completa (es decir, todos los píxeles de la línea completa) comparte el mismo índice de color. En este caso, se transmite el índice de color

- Si el modo es el modo vertical, toda la línea es la misma que la línea superior. En este caso, no se transmite nada. La línea actual copia los índices de color de la línea superior.

- Si el modo es el modo normal, se señala un indicador para cada posición de píxel para indicar si es el mismo con uno de los píxeles de la izquierda y superior. Si no, el propio índice se transmite.

Además, si el píxel es un píxel de escape, se transmite el valor del píxel.

[0123] En otro procedimiento de ejemplo, como se describe, por ejemplo, en el documento de L. Guo, W. Pu, M. Karczewicz, J. Sole, R. Joshi y F. Zou, "RCE4: Results of Test 2 on Palette Mode for Screen Content Coding [RCE4: Resultados de la prueba 2 en el modo de paleta para codificación de contenido de pantalla]", JCTVC-P0198, San José, EE. UU., 9-17 de enero de 2014, se incluye un modo de codificación basada en paleta como un modo CU. El proceso de codificación del segundo procedimiento puede incluir lo siguiente:

- Transmisión de la paleta: se utiliza un esquema de predicción por entrada para codificar la paleta actual basándose en la paleta de la CU izquierda (la CU adyacente a la CU actualmente codificada a la izquierda). Después de eso, se transmiten las entradas no predichas de la paleta.

- Transmisión de valores de píxeles: los píxeles en la CU se codifican en un barrido de trama utilizando los siguientes tres modos:

- "Modo ejecutar": Primero se señala un índice de paleta, seguido de "palette_run" (M). Los siguientes índices de paleta M son los mismos que el índice señalado de paleta que se ha señalado primero.

- "Modo copia de arriba": Se transmite un valor "copy_run" (N) para indicar que los siguientes N índices de paleta son los mismos que sus vecinos superiores, respectivamente.

- "Modo píxel": Primero se transmite un indicador de predicción. El valor del indicador igual a 1 indica predicción residual utilizando el píxel adyacente superior reconstruido a medida que se transmite un predictor. Si el valor de este indicador es 0, el valor de píxel se transmite sin predicción.

[0124] La paleta puede constituir una parte relativamente significativa de los bits para un bloque codificado por paleta (por ejemplo, CU). Por consiguiente, el codificador de vídeo puede predecir una o más entradas de la paleta basándose en una o más entradas de una paleta codificada previamente (por ejemplo, como se indicó anteriormente con respecto a la "transmisión de la paleta").

[0125] En algunos ejemplos, el codificador de vídeo puede generar una lista de predictores de paleta cuando predice entradas de paleta. Por ejemplo, el documento de C. Gisquet, G. Laroche y P. Onno, "AhG10: Palette predictor stuffing [AhG10: Relleno del predictor de paleta]", JCTVC-Q0063 divulga un proceso de ejemplo para determinar los predictores de paleta. En algunos ejemplos, el codificador de vídeo puede usar un vector booleano para indicar si cada elemento en una lista de predictores de paleta se usa (o no) para predecir una o más entradas en la paleta para el bloque que se está codificando actualmente.

[0126] En algunos ejemplos, todos los elementos de la lista de predictores de paleta se obtienen de la paleta codificada previamente (por ejemplo, la paleta codificada con el bloque codificado previamente). Sin embargo, tales paletas pueden estar espacialmente alejadas de la CU actual, lo que puede hacer que la correlación de la paleta sea relativamente débil. En general, la expansión de la tabla de predictores de paleta puede ser útil (por ejemplo, puede proporcionar predictores más precisos, lo que puede resultar en una ganancia de eficiencia). Sin

embargo, la determinación y el uso de una tabla de predictores de paleta relativamente grande da como resultado un vector booleano relativamente más largo.

5 **[0127]** En un ejemplo de codificación de paleta, el codificador de vídeo 20 puede generar un elemento de sintaxis, tal como un indicador "PLT_Mode_flag", que indica si un modo de codificación basada en paleta se utiliza o no para una región particular de una trama de vídeo. Por ejemplo, el PLT_Mode_flag se puede generar en el nivel de fragmento, el nivel de la CU, el nivel de la PU o cualquier otro nivel de una trama de vídeo. Por ejemplo, el codificador de vídeo 20 puede generar el PLT_Mode_flag en el nivel de la CU y señalar el PLT_Mode_flag en un flujo de bits de vídeo codificado. El descodificador de vídeo 30 puede luego analizar el PLT_Mode_flag al descodificar el flujo de bits de vídeo codificado. En este ejemplo, un valor de este PLT_Mode_flag igual a 1 especifica que la CU actual se codifica utilizando una modalidad de paletas. En este caso, el descodificador de vídeo 30 puede aplicar la modalidad de codificación basada en paletas para descodificar la CU. En algunos ejemplos, un elemento de sintaxis puede indicar uno de una pluralidad de diferentes modos de paleta para la CU.

15 **[0128]** Un valor de este PLT_Mode_flag igual a 0 puede especificar que la CU actual se codifica utilizando una modalidad diferente al modo de paleta. Por ejemplo, se puede usar cualquiera de entre varios modos de codificación interpredictiva, intrapredictiva u otros. Cuando un valor de PLT_Mode_flag es 0, se puede transmitir más información para señalar qué modo específico se usa para codificar las CU respectivas, donde dicho modo específico, típicamente, puede ser un modo de codificación HEVC (por ejemplo, intracodificación o intercodificación). El uso de PLT_Mode_flag se describe a modo de ejemplo. Sin embargo, en otros ejemplos, otros elementos de sintaxis, tales como los códigos de múltiples bits, se pueden usar para indicar si el modo de codificación basada en paletas se ha de usar para una CU (o PU en otros ejemplos) o para indicar cuáles, entre una pluralidad de modos, deben ser utilizados.

25 **[0129]** El PLT_Mode_flag u otro elemento de sintaxis también puede transmitirse a un nivel superior. Por ejemplo, el PLT_Mode_flag puede transmitirse a nivel de fragmento. En este caso, un valor del indicador igual a 1 implica que todas las CU en el fragmento se codificarán usando el modo de paleta (lo que significa que no es necesario transmitir información de modo, por ejemplo, para modo de paleta u otros modos a nivel de CU). De manera similar, este indicador se puede señalar a nivel del conjunto de parámetros de imagen (PPS), el conjunto de parámetros de secuencia (SPS) o el conjunto de parámetros de vídeo (VPS). Además, se puede enviar un indicador a uno de estos niveles especificando si el modo de paleta está habilitado o deshabilitado para la imagen, fragmento, etc. en particular, mientras que PLT_Mode_flag indica si el modo de codificación basada en paleta se usa para cada CU. En este caso, si un indicador u otro elemento de sintaxis enviado a nivel de fragmento, PPS, SPS o VPS indica que el modo de codificación por paleta está inhabilitado, en algunos ejemplos, puede que no sea necesario señalar el PLT_Mode_flag para cada CU. De forma alternativa, si un indicador u otro elemento de sintaxis enviado a nivel de fragmento, PPS, SPS o VPS indica que el modo de codificación de paleta está habilitado, el PLT_Mode_flag puede señalizarse adicionalmente para indicar si se ha de usar el modo de codificación basada en paletas para cada CU. De nuevo, como se ha mencionado anteriormente, la aplicación de estas técnicas para indicar la codificación basada en paletas de una CU podría usarse, adicional o de forma alternativa, para indicar la codificación basada en paletas de una PU.

45 **[0130]** Un indicador, como PLT_Mode_flag, también se puede transmitir o inferir condicionalmente o de forma alternativa. Las condiciones para transmitir el PLT_Mode_flag o inferir el indicador puede ser una o más de, como ejemplos, el tamaño de la CU, el tipo de trama, el espacio cromático, el componente cromático, el tamaño de trama, la velocidad de trama, el identificador de capa en la codificación de vídeo ajustable a escala o el identificador de vista en la codificación de múltiples vistas.

50 **[0131]** Ahora se analizarán las técnicas para la generación y transmisión de una paleta. El codificador de vídeo 20 puede configurarse para generar y señalar uno o más elementos de sintaxis y valores que pueden ser utilizados por el descodificador de vídeo 30 para construir y/o reconstruir la paleta utilizada por el codificador de vídeo 20 para codificar un nivel particular de la trama de vídeo (por ejemplo, una CU). En algunos ejemplos, el codificador de vídeo 20 puede indicar o señalar de otro modo una paleta para cada CU. En otros ejemplos, el codificador de vídeo 20 puede indicar o señalar de otro modo una paleta que puede ser compartida entre varias CU.

55 **[0132]** El tamaño de la paleta, por ejemplo, en términos del número de valores de píxeles incluidos, puede ser un valor fijo o puede ser señalado por el codificador de vídeo 20 en un flujo de bits de vídeo codificado. El descodificador de vídeo 30 puede recibir y descodificar la indicación del tamaño de la paleta desde el flujo de bits de vídeo codificado. La señalización se puede separar para diferentes componentes o se puede señalar un solo tamaño para todos los componentes. Los diferentes componentes pueden ser, por ejemplo, componentes de luma y croma. La señalización puede usar códigos unarios o códigos unarios truncados (por ejemplo, que se truncan en un límite máximo del tamaño de la paleta). También se pueden usar los códigos Exponencial-Golomb o Rice-Golomb. En algunos ejemplos, la señalización del tamaño se puede realizar de la siguiente manera: después de señalar una entrada en la paleta, se señala un indicador de "parada". Un valor de este indicador igual a 1 especifica que la entrada actual es la última en la paleta; un valor de este indicador igual a 0 especifica que hay más entradas en la paleta. El indicador de "parada" puede no ser transmitido por el codificador si la paleta ya

construida alcanza el límite máximo del tamaño de la paleta. En algunos ejemplos, el tamaño de la paleta también se puede transmitir o inferir condicionalmente basándose en la información complementaria de la misma manera que se describió anteriormente para la "Transmisión del indicador PLT_Mode_flag".

5 **[0133]** La paleta se puede transmitir por separado para cada componente de color en la CU. Por ejemplo, puede haber una paleta para el componente Y de esta CU, otra paleta para el componente U de esta CU y otra paleta más para el componente V de esta CU. Para la paleta Y, la entrada puede (supuestamente) ser un valor Y representativo en esta CU. Lo mismo se aplica a los componentes U y V. También es posible que la paleta se transmita para todos los componentes de color en la CU. En este ejemplo, la i-ésima entrada en la paleta es triple (por ejemplo, Yi, Ui, Vi). En este caso, la paleta incluye valores para cada uno de los componentes.

15 **[0134]** La predicción de una paleta es un enfoque alternativo a la "transmisión de la paleta" descrita anteriormente. En algunos ejemplos, las técnicas de predicción de la paleta se pueden utilizar junto con las técnicas de señalización de la paleta. Es decir, el codificador de vídeo 20 puede configurarse para señalar elementos de sintaxis que el descodificador de vídeo 30 puede usar para predecir una parte del número total de entradas de la paleta. Además, el codificador de vídeo 20 puede configurarse para señalar explícitamente otra parte de las entradas de la paleta.

20 **[0135]** En un ejemplo de enfoque de predicción de paleta, para cada CU, se transmite un indicador "pred_palette_flag". Un valor de este indicador igual a 1 especifica que la paleta para la CU actual se predecirá a partir de datos pasados y, por lo tanto, no es necesario que la paleta se transmita. Un valor de este indicador igual a 0 significa que la paleta de la CU actual debe ser transmitida. El indicador se puede separar para diferentes componentes de color (por ejemplo, para que se deban transmitir 3 indicadores para una CU en vídeo YUV), o se puede señalar un solo indicador para todos los componentes de color. Por ejemplo, un solo indicador puede indicar si las paletas se transmiten para todos los componentes o si se predecirán las paletas para todos los componentes.

30 **[0136]** En algunos ejemplos, la predicción se puede realizar de la siguiente manera. Si el valor del indicador de predicción es igual a 1, para la CU actual, el codificador de vídeo 20 copia la paleta de una o más de las CU adyacentes ya codificadas. La paleta de las CU adyacentes ya codificadas puede haber sido transmitida o predicha. Por ejemplo, la CU adyacente copiada puede ser la CU adyacente izquierda. En el caso de que la paleta de la CU izquierda no esté disponible (como en el caso de que la CU izquierda no esté codificada utilizando el modo de paleta o la CU actual esté en la primera columna de la imagen), la copia de la paleta puede ser desde la CU encima de la CU actual. La paleta copiada también puede ser una combinación de las paletas de un número de CU adyacentes. Por ejemplo, se pueden aplicar una o más fórmulas, funciones, reglas o similares para generar una paleta basándose en paletas de entre una, o una combinación de una, pluralidad de las CU adyacentes.

40 **[0137]** También es posible que se pueda construir una lista de candidatos y que el codificador de vídeo 20 transmita un índice para indicar la CU candidata desde la cual la CU actual copia la paleta. El descodificador de vídeo 30 puede construir la misma lista de candidatos y, a continuación, usar el índice para seleccionar la paleta de la CU correspondiente, a usar con la CU actual. Por ejemplo, la lista de candidatos puede incluir una CU superior y una CU a la izquierda, en relación con la CU actual a codificar dentro de un fragmento o imagen. En este ejemplo, un indicador u otro elemento de sintaxis puede ser señalado para indicar la selección de candidatos. Por ejemplo, un indicador transmitido igual a 0 significa que la copia es de la CU izquierda, y un indicador transmitido igual a 1 significa que la copia es de la CU superior. El descodificador de vídeo 30 selecciona la paleta que se copiará de la CU adyacente correspondiente y la copia para usarla en la descodificación de la CU actual. La predicción también se puede obtener utilizando los valores de muestra más frecuentes en los vecinos causales de la CU actual.

50 **[0138]** La predicción de paletas también puede ser de entrada. Para cada entrada en la paleta, el codificador de vídeo 20 genera y señala un indicador. Un valor de un indicador igual a 1 para una entrada determinada especifica que se utiliza un valor predicho (por ejemplo, la entrada correspondiente de una CU candidata seleccionada como la CU izquierda) como el valor de esta entrada. Un valor de un indicador igual a 0 especifica que esta entrada no se predice y su valor se transmitirá al descodificador de vídeo 30 desde el codificador de vídeo 20, por ejemplo, señalado en un flujo de bits codificado por el codificador de vídeo 20 para su descodificación posterior por el descodificador de vídeo 30.

60 **[0139]** El valor de "pred_palette_flag", la CU candidata cuya paleta se usa para predecir la paleta de la CU actual o las reglas para construir las candidatas, también se puede transmitir o inferir condicionalmente en base a la información complementaria de la misma manera que se describió anteriormente para la "Transmisión del indicador PLT Mode flag".

65 **[0140]** A continuación, el codificador de vídeo 20 puede generar y señalar una correlación que indica qué respectiva entrada de paleta está asociada con cada píxel en una CU. La i-ésima entrada en la correlación corresponde a la i-ésima posición en la CU. Un valor de la i-ésima entrada igual a 1 especifica que el valor de píxel en esta i-ésima ubicación en la CU es uno de los valores de la paleta, y se transmite además un índice de paleta

para que el descodificador de vídeo 30 pueda reconstruir el valor del píxel (en caso de que solo haya una entrada en la paleta, se puede omitir la transmisión del índice de la paleta). Un valor de la *i*-ésima entrada igual a 0 especifica que el valor de píxel en la *i*-ésima posición en la CU no se encuentra en la paleta y, por lo tanto, el valor de píxel se transmitirá al descodificador de vídeo 30 explícitamente.

5

[0141] Si el valor de píxel en una posición en la CU es un valor en la paleta, se observa que hay una alta probabilidad de que las posiciones adyacentes en la CU tengan el mismo valor de píxel. Entonces, después de codificar un índice de paleta (por ejemplo, *j*, que corresponde al valor de píxel *s*) para una posición, el codificador de vídeo 20 puede transmitir un elemento de sintaxis "run" (ejecución) para indicar el número de valores consecutivos del mismo valor de píxel *s* en la CU antes de que el escaneo alcance un valor de píxel diferente. Por ejemplo, si el siguiente inmediato tiene un valor diferente de *s*, entonces se transmite run=0. Si el siguiente es *s* pero el siguiente no es *s*, entonces run=1.

10

[0142] En el caso de que no se transmita la ejecución (por ejemplo, *Derivación implícita de la ejecución*), el valor de la ejecución puede ser una constante, por ejemplo, 4, 8, 16, etc., o el valor de la ejecución también puede depender de información complementaria. Por ejemplo, el valor de la ejecución puede depender del tamaño del bloque, por ejemplo, la ejecución es igual al ancho del bloque actual, o a la altura del bloque actual, o la mitad del ancho (o media altura) del bloque actual, o una fracción del ancho y la altura del bloque, o un múltiplo de la altura/anchura del bloque. El valor de la ejecución también puede depender del QP, tipo de trama, componente de color, formato de color (por ejemplo, 444, 422, 420) y/o espacio cromático (por ejemplo, YUV, RGB). El valor de la ejecución también puede depender de la dirección de escaneo. En otros ejemplos, el valor de la ejecución puede depender de otros tipos de información complementaria. El valor de la ejecución también se puede señalar usando una sintaxis de alto nivel (por ejemplo, PPS, SPS).

15

20

[0143] En algunos ejemplos, es posible que no sea necesario transmitir la correlación. La ejecución solo puede comenzar en ciertos lugares. Por ejemplo, la ejecución solo puede comenzar al principio de cada fila, o al comienzo de cada *N* filas. La ubicación de inicio puede ser diferente para diferentes direcciones de escaneo. Por ejemplo, si se usa el escaneo vertical, la ejecución solo puede comenzar al principio de una columna o al principio de cada *N* columnas. La ubicación de inicio puede depender de la información complementaria. Por ejemplo, la ubicación de inicio puede ser el punto medio de cada fila, o cada columna, o $1/n$, $2/n$,... $(n-1)/n$ (es decir, fracciones) de cada fila/columna. La ubicación de inicio también puede depender del QP, tipo de trama, componente de color, formato de color (por ejemplo, 444, 422, 420) y/o espacio cromático (por ejemplo, YUV, RGB). En otros ejemplos, la posición de inicio de la ejecución puede depender de otros tipos de información complementaria. La posición de inicio también se puede señalar mediante una sintaxis de alto nivel (por ejemplo, PPS, SPS, etc.).

25

30

35

[0144] También es posible que la obtención de la posición de inicio implícita y la obtención de la ejecución implícita se combinen. Por ejemplo, la ejecución es igual a la distancia entre dos posiciones de inicio adyacentes. En el caso de que el punto de inicio sea el comienzo (es decir, la primera posición) de cada fila, la longitud de la ejecución es una fila.

40

[0145] La dirección de escaneo puede ser vertical u horizontal. Es posible que se transmita un indicador para cada CU para indicar la dirección de escaneo. Los indicadores pueden transmitirse por separado para cada componente o puede transmitirse un solo indicador y la dirección de escaneo indicada se aplica a todos los componentes de color. También es posible que se utilicen otras direcciones de escaneo, como 45 o 135 grados. El orden de escaneo puede ser fijo o puede depender de la información complementaria de la misma manera que se describió anteriormente para la "Transmisión del indicador PLT Mode flag".

45

[0146] Arriba, se explica cómo transmitir una paleta. Una alternativa a los ejemplos descritos anteriormente es construir la paleta sobre la marcha. En este caso, al comienzo de la CU, no hay ninguna entrada en la paleta, y como el codificador de vídeo 20 señala los nuevos valores de los píxeles para las posiciones en la CU, estos valores se incluyen en la paleta. Es decir, el codificador de vídeo 20 puede añadir valores de píxel a una paleta a medida que los valores de píxel se generan y transmiten para posiciones en la CU. Luego, las posiciones posteriores en la CU que tienen los mismos valores pueden referirse a los valores de píxeles en la paleta, por ejemplo, con valores de índice, en lugar de que el codificador de vídeo 20 transmita los valores de píxeles. De manera similar, cuando el descodificador de vídeo 30 recibe un nuevo valor de píxel (por ejemplo, señalado por el codificador) para una posición en la CU, incluye el valor de píxel en la paleta construida por el descodificador de vídeo 30. Cuando las posiciones posteriores de píxeles en la CU tienen valores de píxel que se han añadido a la paleta, el descodificador de vídeo 30 puede recibir información tal como, por ejemplo, valores de índices, que identifican los correspondientes valores de píxeles en la paleta para la reconstrucción de los valores de píxeles en la CU.

50

55

60

[0147] Si se alcanza el tamaño máximo de la paleta, por ejemplo, cuando la paleta se construye dinámicamente sobre la marcha, el codificador y el descodificador comparten el mismo mecanismo para eliminar una entrada de la paleta. Un procedimiento es eliminar la entrada más antigua de la paleta (cola FIFO). Otro procedimiento es eliminar la entrada menos utilizada en la paleta. Otra es ponderar ambos procedimientos (tiempo en la paleta y frecuencia de uso) para decidir la entrada que se va a reemplazar. Como ejemplo, si se elimina una entrada de

65

valor de píxel de la paleta, y el valor de píxel vuelve a aparecer en una posición posterior en la paleta, el codificador puede transmitir el valor de píxel en lugar de incluir una entrada en la paleta. Además, o de forma alternativa, es posible que dicho valor de píxel pueda volver a introducirse en la paleta después de haber sido eliminado, por ejemplo, cuando el codificador y el descodificador escanean las posiciones en la CU.

5

[0148] Esta divulgación también considera la combinación de una señalización de paleta inicial con la obtención sobre la marcha de la paleta. En un ejemplo, la paleta inicial se actualizaría con la codificación de los píxeles. Por ejemplo, al transmitir la paleta inicial, el codificador de vídeo 20 puede agregar valores a la paleta inicial o cambiar los valores en la paleta inicial a medida que se escanean los valores de píxeles de ubicaciones adicionales en la CU. Del mismo modo, al recibir una paleta inicial, el descodificador de vídeo 30 puede agregar valores a la paleta inicial o cambiar los valores en la paleta inicial a medida que se escanean los valores de píxeles de ubicaciones adicionales en la CU. De modo similar, el codificador puede señalar si la CU actual utiliza la transmisión de una paleta completa, o la generación de paletas sobre la marcha, o una combinación de transmisión de una paleta inicial con la actualización de la paleta inicial mediante obtención sobre la marcha. En algunos ejemplos, la paleta inicial puede ser una paleta completa al tamaño máximo de paleta, en cuyo caso los valores de la paleta inicial pueden cambiarse, o una paleta de tamaño reducido, en cuyo caso los valores se agregan a la paleta inicial y, opcionalmente, se cambian los valores en la paleta inicial.

10

15

[0149] Anteriormente, se describió cómo transmitir la correlación identificando el valor de píxel. Junto con el procedimiento descrito anteriormente, la transmisión de la correlación se puede realizar mediante la señalización de la copia de líneas. En un ejemplo, el codificador de vídeo 20 señala la copia de líneas, de manera que el valor de píxel para una entrada es igual al valor de píxel de la entrada de una línea superior (o en la columna de la izquierda si el escaneo es vertical). Luego, se puede señalar la 'ejecución' de las entradas que se copian de la línea. Además, se puede indicar la línea desde la que se copia; se pueden almacenar temporalmente varias líneas superiores para este propósito. Por ejemplo, las cuatro filas anteriores se almacenan y se puede señalar que fila se copia con un código unario truncado u otros códigos, y luego, se puede señalar cuántas entradas de esa fila se copian, es decir, la ejecución. Por lo tanto, en algunos ejemplos, el valor de píxel para una entrada se puede señalar para que sea igual a un valor de píxel de una entrada en una fila inmediatamente superior, o dos o más filas encima de la fila actual.

20

25

30

[0150] En el caso de que no se señalice ninguna ejecución, el valor de la ejecución puede ser constante/fijo o puede depender de la información complementaria (y obtenida por el descodificador) utilizando el procedimiento descrito anteriormente.

35

[0151] También es posible que la correlación no necesite ser transmitida. Por ejemplo, la ejecución puede comenzar solo en ciertas posiciones. La posición de inicio puede ser fija o puede depender de la información complementaria (y obtenida por el descodificador), por lo que se puede omitir la señalización de la posición de inicio. En su lugar, se pueden aplicar una o más de las técnicas descritas anteriormente. La obtención de la posición de inicio implícita y la obtención de la ejecución implícita también se pueden combinar utilizando el mismo procedimiento descrito anteriormente.

40

[0152] Si se usan ambos procedimientos de transmisión de correlación, entonces un indicador u otro elemento de sintaxis puede indicar si el píxel se obtiene de la paleta o de las líneas anteriores, y luego un índice indica la entrada en la paleta o la fila, y finalmente la 'ejecución'.

45

[0153] Esta divulgación describe procedimientos, dispositivos y técnicas para simplificar la codificación en modo paleta y/o para mejorar la eficiencia de la codificación basada en paleta. Las técnicas de esta divulgación se pueden usar en conjunto con otra o por separado para mejorar la eficiencia de la codificación y/o reducir la complejidad del códec. En general, de acuerdo con las técnicas de esta divulgación, un dispositivo de codificación de vídeo puede configurarse para codificar y descodificar datos de vídeo usando uno o más modos de codificación por paleta, en el que los modos de codificación por paleta no incluyen un modo de compartición de paleta.

50

[0154] En un ejemplo de modo de paleta, un indicador, como `palette_share_flag`, puede ser señalado en el flujo de bits para indicar que la paleta para uno o más bloques de datos de vídeo se comparten o se combinan desde la paleta de otro bloque de datos de vídeo. El bloque de datos de vídeo a partir del cual obtener la paleta compartida puede basarse en reglas predeterminadas (por ejemplo, usar la paleta del bloque a la izquierda o encima del bloque actual) o puede indicarse de otra manera en el flujo de bits de vídeo codificado. Como se describe en R. Joshi y J. Xu, "High efficient video coding (HEVC) screen content coding: Draft 2 [Codificación de contenido de pantalla por codificación de vídeo de alta eficiencia (HEVC): borrador 2]", JCTVC-S1005, Sección 7.4.9.6," la semántica de `palette_share_flag` se expresa como "`palette_share_flag[x0][y0]` igual a 1 especifica que la paleta para la unidad de codificación actual se obtiene copiando las primeras entradas `PreviousPaletteSize` de la paleta de predicción. La variable `PreviousPaletteSize` se obtiene como se especifica en la subcláusula 8.4.5.2.8. `palette_share_flag [x0][y0]` igual a 0 especifica que la paleta para la unidad de codificación actual se especifica como una combinación de entradas de la paleta de unidades de codificación anteriores y nuevas entradas de la paleta que se señalizan explícitamente".

55

60

65

[0155] En un ejemplo, cuando el valor de `palette_share_flag` es igual a 1, `palette_share_flag` indica que el bloque actual puede reutilizar la última paleta codificada del bloque codificado previamente. Este procedimiento también se conoce como compartir paleta. Sin embargo, los nuevos resultados de investigación indican que este indicador, junto con el procedimiento de compartir paleta que representa, no es eficaz para mejorar la eficiencia de la codificación, al tiempo que introduce una complejidad adicional de análisis y descodificación.

[0156] Además, algunas redundancias se identifican en el proceso de señalización para un elemento de sintaxis que indica un tipo de ejecución, como `palette_run_type_flag`. Específicamente, cuando el píxel actual es el primer píxel en una línea en un orden de escaneo, y un píxel adyacente al píxel actual y sobre el píxel actual está disponible, el píxel actual no puede estar en el modo "copiar de arriba". El término "el píxel superior está disponible" significa que el vecino superior está dentro del bloque actual para escaneo horizontal o el vecino izquierdo está dentro del bloque para el orden de escaneo vertical si los procedimientos de "copia desde el exterior" no están habilitados. Cuando los procedimientos de "copia desde el exterior" están habilitados, el "píxel superior" puede estar siempre disponible para cada píxel dentro del bloque. Los procedimientos de "copia desde el exterior" de ejemplo se describen en Y.-C. Sun, J. Kim, T.-D. Chuang, Y.-W. Chen, S. Liu, Y.-W. Huang, y S. Lei, "Non-CE6: Cross-CU palette colour index prediction [No CE6: Predicción del índice de color de la paleta entre CU]", JCTVC-S0079 y J. Kim, Y.-C. Sun, S. Liu, T.-D. Chuang, Y.-W. Chen, Y.-W. Huang, y S. Lei, "CE6-related: Enabling copy above mode prediction at the boundary of CU [Relacionado con CE6: Habilitación de la predicción del modo de copia de arriba en el límite de CU]", JCTVC-S0114.

[0157] Si el píxel actual está codificado de acuerdo con el modo "copiar de arriba", entonces el índice del píxel actual es igual al índice del vecino superior del píxel actual. Por el contrario, debido a la regla de que el modo "copiar de arriba" no puede ser seguido inmediatamente por otro modo "copiar de arriba", el vecino superior debe ser el final de una ejecución de "copiar índice". Por lo tanto, la ejecución del "copiar índice" del vecino superior se puede alargar al menos 1 agregando el píxel actual a la ejecución de "copiar índice" en lugar de hacer que el píxel actual sea el primer píxel de una ejecución de "copiar de arriba". Por lo tanto, es posible deshabilitar normativamente el modo "copiar de arriba", si el píxel actual es el primer píxel en una línea en un orden de escaneo. Esto se traduce en un ahorro de bits ya que, para un píxel de este tipo, se puede inferir que el tipo de ejecución es "copiar índice", eliminando así la necesidad de señalar dicho índice.

[0158] Además, la binarización actual para el elemento de sintaxis `palette_num_signalled_entries` está en código unario truncado. El elemento de sintaxis `palette_num_signalled_entries` indica el número de entradas en la paleta actual (por ejemplo, una paleta que se usará para codificar el bloque actual de datos de vídeo) que se señalizan explícitamente. El número de muestras que se señalizan explícitamente se puede determinar por la diferencia entre el número de entradas en la paleta restada del número de entradas en la paleta que se predicen a partir de la paleta de otro bloque de datos de vídeo (incluidas cualesquiera entradas de la paleta que indican el uso de una muestra de escape). En algunos ejemplos, el elemento de sintaxis `palette_num_signalled_entries` puede denominarse el elemento de sintaxis `num_signalled_palette_entries`.

[0159] En algunos ejemplos, la palabra de código utilizada para codificar el valor del elemento de sintaxis `palette_num_signalled_entries` puede ser indeseablemente larga, lo que puede resultar en palabras de código de longitud mayor que 32. Por ejemplo, en HEVC1, todas las palabras de código tienen una longitud de 32 o menos. La misma situación también puede ocurrir al codificar el valor del elemento de sintaxis `palette_predictor_run`. El elemento de sintaxis `palette_predictor_run` especifica el número de ceros que preceden a una entrada distinta de cero en la matriz `palette_entry_reuse_flag`. El indicador `palette_entry_reuse_flag` indica si una entrada de paleta en particular de uno o más paletas utilizadas anteriormente se reutiliza o no para la paleta actual. El valor de `palette_predictor_run` puede variar desde 0 hasta el tamaño máximo de predictor de paleta, inclusive.

[0160] En vista de estos inconvenientes, en un ejemplo de la divulgación, esta divulgación propone que el codificador de vídeo 20 y el descodificador de vídeo 30 estén configurados para realizar un modo de codificación basada en paletas sin técnicas de compartición de paletas. Más específicamente, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar codificación basada en paletas sin usar el elemento de sintaxis `palette_share_flag[x0][y0]`, como se indica a continuación:

[0161] En lugar de utilizar técnicas de compartición de paletas, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para codificar una paleta para usar con uno o más bloques de datos de vídeo usando otras técnicas, como las técnicas de predicción de paletas descritas anteriormente. En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar una predicción de paleta utilizando las siguientes técnicas.

[0162] La figura 4 es un diagrama de bloques que muestra la unidad de codificación basada en paleta 122 del codificador de vídeo 20 con más detalle. La unidad de codificación basada en la paleta 122 puede configurarse para realizar una o más de las técnicas de ejemplo de esta divulgación para la codificación de vídeo basada en la paleta.

[0163] Como se describió anteriormente, la unidad de codificación basada en la paleta 122 puede configurarse para codificar un bloque de datos de vídeo (por ejemplo, una CU o PU) con un modo de codificación basada en la paleta. En un modo de codificación basada en paleta, una paleta puede incluir entradas numeradas por un índice y que representan valores de componentes de color (por ejemplo, RGB, YUV, etc.) o intensidades que pueden usarse para indicar valores de píxeles. La unidad de generación de paleta 203 puede configurarse para recibir valores de píxel 212 para un bloque actual de datos de vídeo y generar una paleta de valores de color para el bloque actual de datos de vídeo. La unidad de generación de paleta 203 puede usar cualquier técnica para generar una paleta para un bloque actual de datos de vídeo, incluidas las técnicas basadas en histogramas analizadas anteriormente. La unidad de generación de paleta 203 puede configurarse para generar una paleta de cualquier tamaño. En un ejemplo, la unidad de generación de paleta 203 puede configurarse para generar 32 entradas de paleta, donde cada entrada de paleta incluye valores de píxel para las componentes Y, Cr y Cb de un píxel. En el ejemplo anterior, se supone que cada entrada de paleta especifica los valores para todos los componentes de color de una muestra (píxel). Sin embargo, los conceptos descritos en el presente documento son aplicables al uso de una paleta independiente para cada componente de color.

[0164] Una vez que una paleta es generada por la unidad de generación de paleta 203, la unidad de correlación 204 puede generar una correlación para el bloque actual de datos de vídeo que indica si un píxel particular en el bloque actual de datos de vídeo puede ser representado o no por una entrada en la paleta generada por la unidad de generación de paleta 203. La unidad de correlación 204 puede producir una correlación 214 que incluye elementos de sintaxis que indican cómo cada píxel usa (o no usa) las entradas de la paleta. Como se analizó anteriormente, en algunos ejemplos, los píxeles de escape no se señalizan con un elemento de sintaxis separado, sino que se pueden indicar con un índice reservado predeterminado en una paleta. Si el valor de un píxel en el bloque actual de datos de vídeo no se encuentra en la paleta, la unidad de correlación 204 puede indicar el uso de un píxel de escape con el índice reservado en la paleta y transmitir explícitamente un valor de píxel para ese píxel en particular. En algunos ejemplos, la unidad de correlación 204 puede predecir el valor de píxel explícito de una de las entradas que se encuentran en la paleta. En algunos otros ejemplos, la unidad de correlación 204 puede cuantificar el píxel y transmitir los valores cuantificados.

[0165] Además de la señalización de los elementos de sintaxis que indica los valores de color utilizados para cada uno de los píxeles en un bloque, la unidad de codificación basada en paleta 122 también puede configurarse para señalar la paleta que se utilizará para un bloque actual de datos de vídeo. De acuerdo con las técnicas de esta divulgación, la unidad de codificación basada en paleta 122 puede configurarse para emplear técnicas de predicción de paleta para reducir la cantidad de datos que se señalizan para indicar los valores de una paleta para un bloque particular de datos de vídeo.

[0166] Como ejemplo de predicción de paleta, como se describe en JCTVC-Q0094, que está disponible a partir del 20 de junio de 2014 en http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q0094-v1.zip, una paleta puede incluir entradas que se copian de una paleta de predicción. Una paleta de predicción puede incluir entradas de paleta de bloques codificados previamente que usan el modo de paleta u otras muestras reconstruidas. Como se muestra en la figura 4, la unidad de codificación basada en paleta 122 puede incluir una memoria intermedia de paleta de predicción 210. La memoria intermedia de paleta de predicción 210 puede configurarse para almacenar un número de entradas de la paleta utilizadas previamente de bloques codificados previamente. Como ejemplo, la memoria intermedia de paleta de predicción 210 puede configurarse como una memoria intermedia de primero en entrar, primero en salir (FIFO) de un tamaño predeterminado. La memoria intermedia de paleta de predicción 210 puede ser de cualquier tamaño. En un ejemplo, la memoria intermedia de paleta de predicción 210 incluye hasta 64 entradas de paleta utilizadas anteriormente.

[0167] En algunos ejemplos, la unidad de codificación basada en paleta 122 puede configurarse para eliminar las entradas en la memoria intermedia de paleta de predicción 210, de modo que todas las entradas de la paleta en la memoria intermedia de paleta de predicción 210 son únicas. Es decir, para cada nueva entrada de paleta que se agregue a la memoria intermedia de paleta de predicción 210, la unidad de codificación basada en paleta 122 puede configurarse para verificar primero que no hay otras entradas idénticas ya almacenadas en la memoria intermedia de paleta de predicción 210. Si no hay entradas idénticas, la nueva entrada de la paleta se agrega a la memoria intermedia de paleta de predicción 210. Si la nueva entrada es un duplicado de una entrada existente, la nueva entrada de la paleta se agrega a la memoria intermedia de paleta de predicción 210 y las entradas duplicadas se eliminan de la memoria intermedia de paleta de predicción 210.

[0168] La unidad de codificación basada en paleta 122 puede incluir una unidad de generación de vectores de predicción binarios 206 que está configurada para generar y señalar un indicador binario (por ejemplo, predictor_palette_entry_reuse_flag), para cada entrada en una paleta para un bloque actual de datos de vídeo generados por la unidad de generación de paletas 203, para indicar si una entrada de la paleta en la memoria intermedia de la paleta de predicción 210 se copia (o se reutiliza) para una de las entradas en la paleta para el bloque actual de datos de vídeo (por ejemplo, indicada por el indicador = 1). Es decir, un indicador con un valor de 1 en el vector de predicción binarios indica que la entrada correspondiente en la memoria intermedia de paleta de predicción 210 se reutiliza para la paleta para el bloque actual, mientras que un indicador con un valor de 0 en el vector de predicción binarios indica que la entrada correspondiente en la memoria intermedia de paleta de

predicción 210 no se reutiliza para la paleta del bloque actual. Además, la unidad de codificación basada en paleta 122 puede configurarse para señalar explícitamente algunos valores para la paleta actual que no se pueden copiar de las entradas en la memoria intermedia de paleta de predicción 210. El número de nuevas entradas también puede ser señalado. A este respecto, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para señalar el número de entradas de paleta señalizadas explícitamente usando el elemento de sintaxis `palette_num_signalled_entries`.

[0169] Cuando se utiliza un modo de codificación basada en paleta que utiliza técnicas de predicción de paleta, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para codificar, entre otros elementos de sintaxis, un elemento de sintaxis que indica el número de entradas de paleta que se señalizan explícitamente para una paleta actual para ser usada para codificar el bloque actual de datos de vídeo (por ejemplo, `palette_num_signalled_entries`). Esta divulgación propone técnicas para mejorar la eficiencia de la codificación o restringir la longitud de la palabra de código cuando se codifica un elemento de sintaxis de este tipo.

[0170] En un ejemplo de la divulgación, la unidad de codificación basada en paleta 122 se puede configurar para codificar el primer bin de un elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente, como el elemento de sintaxis `palette_num_signalled_entries`, utilizando un contexto CABAC. La unidad de codificación basada en paleta 122 puede codificar otros bin de `palette_num_signalled_entries` utilizando otras técnicas de codificación. En otro ejemplo de la divulgación, la unidad de codificación basada en paleta 122 puede configurarse para usar más de un contexto para codificar el primer bin del elemento de sintaxis `palette_num_signalled_entries`. En un ejemplo, la unidad de codificación basada en paleta 122 se puede configurar para determinar los contextos basándose en el tamaño de bloque del bloque de vídeo actual que se está codificando y/o basándose en el valor de otros elementos de sintaxis.

[0171] De acuerdo con un ejemplo de la divulgación, la unidad de codificación basada en paleta 122 puede configurarse para determinar un primer bin de un primer elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente. El codificador de vídeo 20 también puede configurarse para codificar un flujo de bits que incluya el primer elemento de sintaxis. El flujo de bits también puede no incluir un segundo elemento de sintaxis que indica un modo de compartición de paleta. En algunos ejemplos, la unidad de codificación basada en paleta 122 puede configurarse para codificar el primer bin del primer elemento de sintaxis utilizando una codificación aritmética binaria adaptada al contexto. En otros ejemplos, la unidad de codificación basada en la paleta 122 puede configurarse para codificar el primer bin del primer elemento de sintaxis utilizando uno o más contextos. En algunos ejemplos de uso de uno o más contextos, el uno o más contextos pueden basarse en al menos uno de un número predicho de entradas de codificación de la paleta o un tamaño de bloque.

[0172] En otro ejemplo de la divulgación, para evitar que la longitud de la palabra de código `palette_num_signalled_entries` crezca más de 32 bits, se propone que se realicen cambios semánticos normativos a las técnicas actuales de codificación de paletas (por ejemplo, R. Joshi y J. Xu, "High efficient video coding (HEVC) screen content coding: Draft 2 [Codificación de contenido de pantalla por codificación de vídeo de alta eficiencia (HEVC): Borrador 2]", JCTVC-S1005). Por ejemplo, los valores factibles de un elemento de sintaxis que especifica el tamaño máximo permitido de la paleta, como `palette_max_size`, y un elemento de sintaxis que especifica el tamaño máximo de la paleta de predicción, como `palette_max_predictor_size`, pueden estar limitados por un umbral. Dicho umbral puede determinarse y almacenarse en una memoria (por ejemplo, la memoria de datos de vídeo 98 en la figura 2 o la memoria de datos de vídeo 148 en la figura 3) a la que se puede acceder mediante la unidad de codificación basada en paleta 122. Específicamente, para `palette_max_size`, el valor puede ser cualquier valor de 0 a T1, inclusive, donde T1 es el umbral. Cuando no está presente, la unidad de codificación basada en la paleta 122 se puede configurar para inferir que el valor de `palette_max_size` sea 0. Además, para `palette_max_predictor_size`, el valor puede ser cualquier valor de 0 a T2, inclusive, donde T2 es el umbral. Cuando no está presente, la unidad de codificación basada en paleta 122 se puede configurar para inferir que el valor de `palette_max_predictor_size` sea 0.

[0173] En un ejemplo, T1 es igual a 4096 y T2 es igual a 8192. En otro ejemplo, T1 es igual a 4095 y T2 es igual a 4095. En aún otro ejemplo, T1 es igual a 4095 y T2 es igual a 8191.

[0174] Como otro ejemplo, esta divulgación propone que el valor de `palette_max_size` sea igual al número de píxeles en la unidad de codificación de tamaño más grande. Dicho valor puede ser predeterminado y almacenado en una memoria accesible por la unidad de codificación basada en paleta 122. En algunos ejemplos, el valor de `palette_max_predictor_size` puede ser menor o igual que $K * \text{palette_max_size}$, donde K es una constante positiva. En algunos ejemplos, $K = 2$.

[0175] En otro ejemplo, la unidad de codificación basada en paleta 122 (por ejemplo, utilizando la unidad de compresión vectorial binaria 209 u otro componente estructural del codificador de vídeo 20, como la unidad de codificación por entropía 118) puede configurarse para codificar el valor del elemento de sintaxis `palette_num_signalled_entries` usando una o más técnicas de codificación de la familia de códigos Golomb (por ejemplo, código Golomb-Rice, código Exponencial Golomb, código de Rice truncado, código unario, etc.). En un ejemplo de la divulgación, la unidad de codificación basada en paleta 122 está configurada para codificar el valor

del elemento de sintaxis `palette_num_signalled_entries` usando un código exponencial Golomb de orden 0. En otro ejemplo de la divulgación, la unidad de codificación basada en paleta 122 está configurada para codificar el valor del elemento de sintaxis `palette_num_signalled_entries` usando una concatenación de un código de Rice truncado (TR) y un código exponencial Golomb como el que se usa en la codificación de coeficientes para codificar el elemento de sintaxis `coeff_abs_level_remaining` en HEVC1.

[0176] A continuación se muestra un ejemplo de una concatenación de un código TR y un código exponencial Golomb para el parámetro Golomb Rice de 0:

Símbolo	binarización
0	0
1	10
2	110
3	1110
[4, 5]	11110x
[6, 9]	111110xx
[10, 17]	1111110xxx
...	

Aquí x puede tener un valor de 0 o 1. De manera similar, la tabla a continuación muestra un ejemplo de una binarización concatenada utilizada en la codificación del elemento de sintaxis `paletteRun`. Esta es una concatenación de un código de Rice truncado y de un código exponencial Golomb truncado de orden 0 para un valor de ejecución máximo de 7.

símbolo	binarización
0	0
1	10
[2, 3]	110x
[4, 7]	111xx

Aquí x puede tener un valor de 0 o 1.

[0177] El uso de uno o más códigos de Golomb (por ejemplo, como un código exponencial Golomb o una concatenación de código de TR y un código exponencial Golomb) para codificar el elemento de sintaxis `palette_num_signalled_entries` proporciona un beneficio en comparación con las técnicas anteriores para codificar el valor del elemento de sintaxis `palette_num_signalled_entries`. Las técnicas anteriores para codificar el valor del elemento de sintaxis `palette_num_signalled_entries` utilizaban un código unario. El uso de un código unario dio como resultado que la longitud codificada del elemento de sintaxis `palette_num_signalled_entries` fuera superior a 32 bits en algunas circunstancias. Al usar uno o más códigos de Golomb para codificar el elemento de sintaxis `palette_num_signalled_entries`, las técnicas de esta divulgación permiten que la unidad de codificación basada en paleta 122 codifique el valor del elemento de sintaxis `palette_num_signalled_entries` en una materia que mantiene la longitud codificada en o inferior algún número predeterminado de bits (por ejemplo, 32 bits).

[0178] En otro ejemplo, la unidad de codificación basada en paleta 122 se puede configurar para codificar el valor del elemento de sintaxis `palette_num_signalled_entries` usando una versión truncada de una familia de códigos Golomb (por ejemplo, código Golomb Rice truncado, código exponencial Golomb truncado, código truncado de Rice truncado, código unario truncado, etc.). En otro ejemplo de la divulgación, la unidad de codificación basada en paleta 122 se puede configurar para codificar el valor del elemento de sintaxis `paletteRun`. En otro ejemplo, la unidad de codificación basada en paleta 122 se puede configurar para codificar el valor del elemento de sintaxis `palette_num_signalled_entries` usando el procedimiento utilizado para codificar el elemento de sintaxis `coeff_abs_level_remaining` en la codificación de coeficientes (por ejemplo, la concatenación de código Rice truncado (TR) y exponencial Golomb). De acuerdo con este ejemplo, se prefiere que el parámetro TR sea 0. En cada uno de estos ejemplos, el código truncado en particular se elige de tal manera que la longitud codificada del elemento de sintaxis `palette_num_signalled_entries` se mantenga en o por debajo de 32 bits.

[0179] En otro ejemplo, se propone imponer una restricción en el flujo de bits que `palette_num_signalled_entries` es igual a la cantidad de píxeles en el bloque. Es decir, la unidad de codificación basada en paleta 122 se puede configurar para limitar el posible valor del elemento de sintaxis `palette_num_signalled_entries` por el número de

píxeles en el bloque codificado actualmente. En otro ejemplo, la unidad de codificación basada en paleta 122 puede configurarse para limitar el posible valor de `palette_num_signalled_entries` por el número de píxeles en el bloque más grande posible de una imagen particular (por ejemplo, el gran tamaño de bloque definido por una norma de codificación de vídeo particular).

[0180] En otro ejemplo, la unidad de codificación basada en paleta 122 puede configurarse para omitir la señalización de un elemento de sintaxis que indica un tipo de ejecución, como `palette_run_type_flag`, si el píxel actual es el primer píxel en la línea en el orden de escaneo y el píxel adyacente al píxel actual por encima del píxel actual está disponible. En un ejemplo, la unidad de codificación basada en paleta 122 puede configurarse para determinar que un píxel actual es un primer píxel en una línea en un orden de escaneo. La unidad de codificación basada en paleta 122 puede determinar además que un píxel adyacente situado encima del píxel actual está disponible. En respuesta a la determinación de que el píxel actual es el primer píxel en la línea en el orden de escaneo y a la determinación de que el píxel adyacente situado sobre el píxel actual está disponible, la unidad de codificación basada en paleta 122 puede configurarse además para omitir la codificación de un primer elemento de sintaxis en un flujo de bits, en la que el primer elemento de sintaxis indica un tipo de ejecución y codifica un resto del flujo de bits.

[0181] Volviendo a la figura 4 y las técnicas de predicción de paleta de esta divulgación, en la Solicitud de EE. UU. Nº 14/667.411, presentada el 24 de marzo de 2015, publicada como Publicación de Patente de EE. UU. Nº 2015/0281728, se propusieron un procedimiento de señalización basada en árbol binario y procedimientos de señalización basados en la posición final para la codificación del vector predictor binario de la paleta. En la Solicitud Provisional de EE. UU. No. 62/002.741, presentada el 23 de mayo de 2014, se propuso un procedimiento de señalización basado en grupo. Esta divulgación propone técnicas adicionales para generar, codificar y decodificar el vector de predicción binario.

[0182] Algunos ejemplos descritos en el presente documento se relacionan con procedimientos para codificar el vector de predicción de paleta para mejorar la eficiencia de la codificación. Por ejemplo, suponga que el vector de predicción binario generado por la unidad de generación del vector de predicción binario 206 se denota por:

$$\mathbf{b} = [b_0, b_1, \dots, b_{N-1}], N \geq 0, b_i \in \{0, 1\}, 0 \leq i < N$$

En la ecuación anterior, $b_i \in \{0, 1\}$, $0 \leq i < N$ denota un indicador de predicción (también denominado indicador binario o indicador de predicción binario). Si $N=0$, $\mathbf{b} = \varnothing$ (es decir, \mathbf{b} es el vector vacío), que no necesita ser señalado. Por lo tanto, en la siguiente descripción, se puede suponer que $N > 0$.

[0183] La figura 5 muestra un ejemplo de una memoria intermedia de paleta de predicción 210 y una paleta 220 actual. Como puede verse en la figura 5, la paleta 220 actual reutiliza los valores de píxel de la memoria intermedia de la paleta de predicción 210 asociada con los índices de entrada 1, 2, 5 y 9. Como tal, un vector predictor binario producido por la unidad de generación de vectores de predicción binarios 206 de la figura 4 sería $\mathbf{b} = [110010001000]$. Como se puede ver en este ejemplo, el vector de predicción binario \mathbf{b} incluye indicadores con un valor de 1 correspondiente a los índices 1^o, 2^o, 5^o y 9^o en la memoria intermedia de paleta de predicción 210. Es decir, la 1^a, 2^a, 5^a y 9^a entradas en la memoria intermedia de paleta de predicción 210 son las únicas entradas reutilizadas para paleta actual 220. Para los índices de entrada 5-8 en la paleta actual 220, la unidad de codificación basada en paleta 122 puede configurarse para señalar valores de entrada de la paleta en el flujo de bits de vídeo codificado (por ejemplo, utilizando señalización explícita u otra técnica de predicción).

[0184] De acuerdo con una o más técnicas de esta divulgación, el codificador de vídeo 20 puede configurarse para codificar o codificar, en general, el vector \mathbf{b} predictor binario para reducir la cantidad de datos necesarios para señalar una paleta en el flujo de bits de vídeo codificado. Como se muestra en la figura 4, la unidad de compresión del vector de predicción binario 209 puede configurarse para generar y señalar el vector de predicción binario 215 codificado. Sin embargo, debe entenderse que las técnicas de compresión del vector de predicción binario de esta divulgación pueden implementarse en otras estructuras del codificador de vídeo 20, incluida la unidad de codificación de entropía 118 en la figura 2.

[0185] En un ejemplo de la divulgación, la unidad de compresión del vector de predicción binario 209 puede configurarse para codificar el vector de predicción binario utilizando técnicas de codificación basadas en longitud de ejecución. Por ejemplo, la unidad de compresión del vector de predicción binario 209 puede configurarse para codificar el vector de predicción binario mediante la señalización del número de '0' consecutivos entre '1' en el vector de predicción binario utilizando un código exponencial-Golomb. Como ejemplo, supongamos nuevamente que $\mathbf{b} = [110010001000]$. En este ejemplo, como se muestra en la figura 6, el vector de predicción binario (es decir, \mathbf{b}) se puede expresar como: 'cero 0 consecutivos'-1'-cero 0 consecutivos'-1'- 'dos 0 consecutivos'-1'-'tres 0 consecutivos'-1'-y 'cuatro 0 consecutivos'. Como se sabe que $b_i \in \{0, 1\}$, a excepción del último grupo '0 consecutivo', cada grupo '0 consecutivo' debe ir seguido de un '1'. Por lo tanto, la unidad de compresión de vector de predicción binario 209 puede utilizar técnicas de codificación por longitud de ejecución basada en cero para representar el vector de predicción binario \mathbf{b} como 'cero 0 consecutivos'-cero 0 consecutivos'-dos 0 consecutivos'-

'tres 0 consecutivos'- 'cuatro 0 consecutivos', que se puede expresar como la secuencia de longitud de ejecución '0-0-2-3-4'.

[0186] De acuerdo con uno o más ejemplos de esta divulgación relacionados con la señalización basada en longitud de ejecución, para codificar la secuencia de longitud de ejecución se puede utilizar un código Golomb-Rice, un código Exponencial-Golomb de cualquier orden, un código Exponencial-Golomb truncado, un código Rice truncado o cualquier otra binarización, incluyendo binarizaciones truncadas. En un ejemplo, la unidad de compresión del vector de predicción binario 209 utiliza un código de Exponencial-Golomb de orden 0 como la técnica de codificación de longitud de ejecución.

[0187] Para la binarización truncada, maxsymbol puede ser el valor máximo posible de la ejecución en función de la posición de '1' en el vector binario y el tamaño del vector binario, ya que, al pasar al final del vector binario, el valor de la ejecución máxima posible se reduce del tamaño del vector a 0 dependiendo de la posición dentro del vector. Por ejemplo, maxsymbol puede ser la longitud del vector binario o la longitud del vector binario menos la posición del '1' desde el cual se cuenta la ejecución. En otras palabras, es la longitud restante medida desde el final del vector binario. Para el ejemplo anterior con el vector binario \mathbf{b} de un tamaño particular, por ejemplo, 13, la secuencia de longitud de ejecución '0-0-2-3-4' puede codificarse con la binarización truncada 0[13]-0[12]-2[11]-3[8]-4[4]', donde maxsymbol se encuentra entre corchetes.

[0188] Además, en algunos ejemplos, la binarización puede depender de la posición o el índice del elemento (0 o 1) en el vector binario. Como ejemplo particular, si la posición es más pequeña que un cierto umbral, se usa un tipo de binarización; de lo contrario, se aplica otro tipo de binarización. En algunos ejemplos, el tipo de binarización puede ser un código de binarización diferente, o la misma familia de códigos, pero con un orden diferente, como el código Exponencial-Golomb.

[0189] En un ejemplo, el umbral puede ser la longitud de la paleta del bloque anterior o del bloque codificado por paleta anterior. En otro ejemplo, el umbral puede fijarse a un valor predeterminado o señalizarse por bloque, fragmento, imagen o en cualquier otro lugar. Se debe reconocer que una técnica correspondiente se puede usar opcionalmente para definir un contexto CABAC para codificar los valores de ejecución. Además, la unidad de codificación basada en paleta 122 (ver la figura 2) puede configurarse para detener la señalización de longitud de ejecución cuando el número de elementos '1' señalizados (es decir, el número de entradas de la paleta desde la memoria intermedia de paleta de predicción 210 indicado como reutilizado por la paleta actual 220) alcanza un número máximo posible. En algunos ejemplos, el número máximo posible es el tamaño máximo posible de la paleta.

[0190] Algunos ejemplos de esta divulgación se refieren a la codificación de la posición final de la secuencia de la longitud de ejecución que indica el vector de predicción binario \mathbf{b} . En uno o más ejemplos de esta divulgación, la unidad de compresión del vector de predicción binario 209 puede configurarse para codificar el vector de predicción binario \mathbf{b} utilizando una longitud de ejecución reservada L para codificar la posición final del vector de predicción binario. En un ejemplo, $L=1$ se utiliza como la longitud de ejecución reservada. En el codificador de vídeo 20, si la longitud de ejecución es igual o mayor que L , la unidad de compresión del vector de predicción binario 209 está configurada para agregar 1 a la longitud de ejecución. Si la longitud de ejecución real es menor que L , la unidad de compresión del vector de predicción binario 209 está configurada para señalar la longitud de ejecución tal como está. La unidad de compresión del vector de predicción binario 209 puede señalar la longitud de ejecución de la posición final con la longitud de ejecución reservada L .

[0191] Del mismo modo, en el descodificador de vídeo 30, si el valor descodificado de una longitud de ejecución es mayor que L , 1 se resta de la longitud de ejecución real. Si el valor descodificado o una longitud de ejecución es menor que L , el valor descodificado se utiliza como la longitud de ejecución real. Si el valor descodificado es igual a L , las posiciones restantes en el vector de predicción binario \mathbf{b} son todas 0. Por lo tanto, si el valor descodificado es igual a L , no es necesaria más señalización de ejecución.

[0192] Utilizando el mismo ejemplo anterior (es decir, $\mathbf{b} = [110010001000]$) y suponiendo que $L=1$, la unidad de compresión del vector de predicción binario 209 está configurada para señalar la secuencia de longitud de ejecución '0-0-2-3-4' de la figura 6 como '0-0-3-4-1'. Luego, aplicando las reglas anteriores, el descodificador de vídeo 30 se puede configurar para recuperar la secuencia de longitud de ejecución como '0-0-2-3-final'. Es decir, el primer valor de longitud de ejecución de 0 se descodifica como 0 y la siguiente secuencia de longitud de ejecución de 0 se descodifica como 0, ya que ambas secuencias de longitud de ejecución 0 son menores que el valor de longitud de ejecución reservada de $L = 1$. La siguiente secuencia de longitud de ejecución es 3, y como tal, el descodificador de vídeo 30 se configurará para restar 1 del valor de 3 para obtener 2, porque el valor recibido de 3 es mayor que el valor de longitud de ejecución reservada de $L=1$. Del mismo modo, el descodificador de vídeo 30 se configurará para restar 1 del valor recibido de 4 para obtener 3 para la siguiente secuencia de longitud de ejecución, porque el valor recibido de 4 es mayor que el valor de la longitud de ejecución reservada de $L=1$. Finalmente, el último valor recibido de longitud de ejecución 1 es igual al valor de longitud de ejecución reservada de $L=1$. Por consiguiente, el descodificador de vídeo 30 puede determinar que no hay otros valores de '1' presentes en el vector de predicción binario.

[0193] La figura 7 es un diagrama de bloques que ilustra una unidad de descodificación basada en paleta 165 de ejemplo de un descodificador de vídeo 30. La unidad de descodificación basada en paleta 165 puede configurarse para funcionar de manera recíproca con la unidad de codificación basada en paleta 122 de la figura 4. La unidad de descodificación basada en paleta 165 puede configurarse para recibir una correlación 312 que indica, para cada píxel en un bloque actual, si las entradas para una paleta se usarán o no para los píxeles en el bloque actual. Además, la correlación 312 puede indicar además qué entradas de la paleta se utilizarán para un píxel dado. La unidad de correlación 302 puede descodificar el bloque actual de datos de vídeo usando la correlación 312 y una paleta generada por la unidad de generación de paletas 304 para producir datos de vídeo descodificados 314.

[0194] De acuerdo con las técnicas de esta divulgación, la unidad de descodificación basada en paleta 165 también puede recibir un vector de predicción binario codificado 316. Como se analizó anteriormente, el vector de predicción binario 316 puede codificarse usando una técnica de codificación de longitud de ejecución que codifica una secuencia de longitud de ejecución que indica una ejecución de valores cero en el vector de predicción binario. La unidad de descompresión del vector de predicción binario 306 puede configurarse para descodificar el vector de predicción binario codificado 316 usando cualquier combinación de las técnicas de codificación de longitud de ejecución descritas anteriormente con referencia a las figuras 4-6. Una vez que se recupera un vector de predicción binario mediante la unidad de descompresión del vector de predicción binario 306, la unidad de generación de paleta 304 puede generar una paleta para el bloque actual de datos de vídeo basándose en el vector de predicción binario y las entradas de la paleta utilizadas previamente almacenadas en la memoria intermedia de paletas de predicción 310. La unidad de descodificación basada en paleta 165 puede configurarse para almacenar las entradas de la paleta usadas previamente en la memoria intermedia de paleta de predicción 310 de la misma manera que la unidad de codificación basada en paleta 122 (ver figura 2) almacenó las entradas de la paleta previamente utilizadas en la memoria intermedia de paleta de predicción 210.

[0195] En un ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse para descodificar el primer bin de un elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente, como el elemento de sintaxis `palette_num_signalled_entries`, utilizando una CABAC de contexto. La unidad de descodificación basada en paleta 165 puede descodificar otros bin de las entradas de `palette_num_signalled_entries` utilizando otras técnicas de descodificación. En otro ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse para usar más de un contexto para descodificar el primer bin del elemento de sintaxis `palette_num_signalled_entries`. En un ejemplo, la unidad de descodificación basada en paleta 165 puede configurarse para determinar los contextos basándose en el tamaño de bloque del bloque de vídeo actual que se está descodificando y/o basándose en el valor de otros elementos de sintaxis.

[0196] De acuerdo con un ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse para determinar un primer bin de un primer elemento de sintaxis que indica un número de entradas en una paleta actual que están señalizadas explícitamente. El descodificador de vídeo 30 también puede configurarse para descodificar un flujo de bits que incluye el primer elemento de sintaxis. El flujo de bits también puede no incluir un segundo elemento de sintaxis que indica un modo de compartición de paleta. En algunos ejemplos, la unidad de descodificación basada en paleta 165 puede configurarse para descodificar el primer bin del primer elemento de sintaxis usando una codificación aritmética binaria adaptada al contexto. En otros ejemplos, la unidad de descodificación basada en paleta 165 puede configurarse para descodificar el primer bin del primer elemento de sintaxis utilizando uno o más contextos. En algunos ejemplos de uso de uno o más contextos, el uno o más contextos pueden basarse en al menos uno de un número predicho de entradas de codificación de la paleta o un tamaño de bloque.

[0197] En otro ejemplo de la divulgación, para evitar que la longitud de la palabra de código de `palette_num_signalled_entries` crezca más de 32 bits, se propone que se realicen cambios semánticos normativos a las técnicas actuales de codificación de paletas. Por ejemplo, los valores factibles de un elemento de sintaxis que especifica el tamaño máximo permitido de la paleta, como `palette_max_size`, y un elemento de sintaxis que especifica el tamaño máximo de la paleta de predicción, como `palette_max_predictor_size`, pueden estar limitados por un umbral. Dicho umbral puede predeterminarse y almacenarse en una memoria (por ejemplo, la memoria de datos de vídeo 148 en la figura 3) que es accesible por la unidad de descodificación basada en paleta 165. Específicamente, para `palette_max_size`, el valor puede ser cualquier valor de 0 a T1, inclusive, donde T1 es el umbral. Cuando no está presente, la unidad de descodificación basada en paleta 165 puede configurarse para inferir que el valor de `palette_max_size` sea 0. Además, para `palette_max_predictor_size`, el valor puede ser cualquier valor de 0 a T2, inclusive, donde T2 es el umbral. Cuando no está presente, la unidad de descodificación basada en paleta 165 puede configurarse para inferir que el valor de `palette_max_predictor-size` sea 0.

[0198] En un ejemplo, T1 es igual a 4096 y T2 es igual a 8192. En otro ejemplo, T1 es igual a 4095 y T2 es igual a 4095. En aún otro ejemplo, T1 es igual a 4095 y T2 es igual a 8191.

[0199] Como otro ejemplo, esta divulgación propone que el valor de `palette_max_size` sea igual al número de píxeles en la unidad de codificación de máximo tamaño. Dicho valor puede determinarse y almacenarse en una memoria accesible por la unidad de descodificación basada en paleta 165. En algunos ejemplos, el valor de `palette_max_predictor_size` puede ser menor o igual que $K * \text{palette_max_size}$, donde K es una constante positiva. En algunos ejemplos, $K = 2$.

[0200] En otro ejemplo, la unidad de descodificación basada en paleta 165 de la figura 3 (por ejemplo, utilizando la unidad de descompresión vectorial de predicción binario 306 u otro componente estructural del descodificador de vídeo 30, como la unidad de descodificación por entropía 150 de la figura 3) puede configurarse para descodificar el valor del elemento de sintaxis `palette_num_signalled_entries` usando una o más técnicas de descodificación de la familia de códigos Golomb (por ejemplo, código Golomb-Rice, código Exponencial Golomb, código de Rice truncado, código unario, etc.). En un ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 está configurada para descodificar el valor del elemento de sintaxis `palette_num_signalled_entries` usando una concatenación de código de Rice truncado y Exponencial Golomb.

[0201] En otro ejemplo, la unidad de descodificación basada en paleta 165 puede configurarse para descodificar el valor del elemento de sintaxis `palette_num_signalled_entries` usando una versión truncada de una familia de códigos de Golomb (por ejemplo, código Golomb Rice truncado, código Exponencial Golomb truncado, código truncado de Rice truncado, código unario truncado, etc.). En otro ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse para descodificar el valor del elemento de sintaxis `palette_num_signalled_entries` usando el mismo código usado para codificar el elemento de sintaxis `paletteRun`. En otro ejemplo, la unidad de descodificación basada en paleta 165 puede configurarse para descodificar el valor del elemento de sintaxis `palette_num_signalled_entries` utilizando el procedimiento de descodificación del elemento de sintaxis `coeff_abs_level_remaining` en la descodificación de coeficientes (por ejemplo, concatenación de código de Rice truncado (TR) y el exponencial Golomb). De acuerdo con este ejemplo, se prefiere que el parámetro TR sea 0.

[0202] En otro ejemplo, se propone imponer una restricción en el flujo de bits que `palette_num_signalled_entries` es igual al número de píxeles en el bloque. Es decir, la unidad de descodificación basada en paleta 165 puede configurarse para limitar el valor posible del elemento de sintaxis `palette_num_signalled_entries` por el número de píxeles en el bloque codificado actualmente. En otro ejemplo, la unidad de descodificación basada en paleta 165 puede configurarse para limitar el posible valor de `palette_num_signalled_entries` por el número de píxeles en el bloque más grande posible de una imagen en particular (por ejemplo, el tamaño de bloque más grande definido por una norma de codificación de vídeo en particular).

[0203] En otro ejemplo, la unidad de descodificación basada en paleta 165 puede configurarse para inferir el valor de un elemento de sintaxis que indica un tipo de ejecución, como `palette_run_type_flag`, si el píxel actual es el primer píxel en la línea en el orden de escaneo y el píxel adyacente al píxel actual sobre el píxel actual está disponible. En un ejemplo, la unidad de descodificación basada en paleta 165 puede configurarse para determinar que un píxel actual es un primer píxel en una línea en un orden de escaneo. La unidad de descodificación basada en paleta 165 puede determinar además que está disponible un píxel adyacente situado encima del píxel actual. En respuesta a la determinación de que el píxel actual es el primer píxel en la línea en el orden de escaneo y al determinar que el píxel adyacente situado sobre el píxel actual está disponible, la unidad de descodificación basada en paleta 165 puede configurarse aún más para inferir el valor del primer elemento de sintaxis en un flujo de bits, en el que el primer elemento de sintaxis indica un tipo de ejecución y codifica un resto del flujo de bits.

[0204] La figura 8 es un diagrama de flujo que ilustra un procedimiento de codificación de vídeo a modo de ejemplo de acuerdo con las técnicas de esta divulgación. Las técnicas de la figura 8 puede implementarse mediante una o más estructuras de hardware del codificador de vídeo 20, que incluyen la unidad de codificación basada en paleta 122 y/o la unidad de codificación por entropía 118 (ver la figura 2).

[0205] En un ejemplo de la divulgación, el codificador de vídeo 20 puede configurarse para codificar un bloque de datos de vídeo usando un modo de codificación basada en paleta y una paleta (800), y generar una pluralidad de elementos de sintaxis que son indicativos de la paleta que se usó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado (802). El codificador de vídeo 20 puede configurarse adicionalmente para codificar el primer elemento de sintaxis utilizando uno o más códigos de Golomb, de modo que la longitud del primer elemento de sintaxis codificado sea menor o igual a un número predeterminado de bits (804), e incluye la pluralidad de elementos de sintaxis en un flujo de bits de vídeo codificado (806).

[0206] En un ejemplo de la divulgación, el primer elemento de sintaxis es un elemento de sintaxis `palette_num_signalled_entries`. En otro ejemplo de la divulgación, la pluralidad de elementos de sintaxis incluye los valores de paleta indicados como señalizados explícitamente por el primer elemento de sintaxis.

[0207] En un ejemplo de la divulgación, el número máximo predeterminado de bits es 32, y el uno o más códigos de Golomb es un código exponencial Golomb de orden 0. En otro ejemplo de la divulgación, el número máximo predeterminado de bits es 32, y el uno o más códigos de Golomb es una concatenación de un código de Rice truncado y un código exponencial Golomb.

5

[0208] En otro ejemplo de la divulgación, se define un valor máximo del primer elemento de sintaxis en relación con un segundo elemento de sintaxis que indica un tamaño máximo de la paleta y un tercer elemento de sintaxis que indica un tamaño máximo de un predictor de paleta. En este ejemplo, el codificador de vídeo 20 puede configurarse además para definir el segundo elemento de sintaxis como un valor de 0 a un primer umbral, y definir el tercer elemento de sintaxis como un valor de 0 a un segundo umbral. En un ejemplo, el primer umbral es uno de 4095 o 4096 y el segundo umbral es uno de 4095, 8191 u 8192.

10

[0209] En otro ejemplo de la divulgación, se define un valor máximo del primer elemento de sintaxis en relación con un segundo elemento de sintaxis que indica un tamaño máximo de la paleta y un tercer elemento de sintaxis que indica un tamaño máximo de un predictor de paleta. En este ejemplo, el codificador de vídeo 20 puede configurarse además para definir que el segundo elemento de sintaxis sea menor o igual a un número de píxeles en el bloque más grande posible en el flujo de bits de vídeo codificado, y definir que el tercer elemento de sintaxis sea menor o igual a $K \cdot$ un valor del segundo elemento de sintaxis, donde K es una constante positiva. En un ejemplo, K es 2.

15

20

[0210] En otro ejemplo de la divulgación, el codificador de vídeo 20 puede configurarse además para señalar un elemento de sintaxis que indica un tipo de ejecución de paleta en el caso de que un píxel actual no sea un primer píxel en un orden de escaneo, y no señalar el elemento de sintaxis que indica un tipo de ejecución de paleta en el caso de que el píxel actual sea el primer píxel en el orden de escaneo y el píxel/muestra anterior esté disponible.

25

[0211] La figura 9 es un diagrama de flujo que ilustra un procedimiento de descodificación de vídeo de ejemplo de acuerdo con las técnicas de esta divulgación. Las técnicas de la figura 9 puede implementarse mediante una o más estructuras de hardware del descodificador de vídeo 30, que incluyen la unidad de descodificación basada en paleta 165 y/o la unidad de descodificación por entropía 150 (ver figura 3).

30

[0212] En un ejemplo de la divulgación, el descodificador de vídeo 30 puede configurarse para recibir un bloque de datos de vídeo en un flujo de bits de vídeo codificado, habiéndose codificado el bloque de datos de vídeo usando un modo de codificación basada en paleta (900), y recibir una pluralidad de elementos de sintaxis que son indicativos de una paleta que se usó para codificar el bloque de datos de vídeo, la pluralidad de elementos de sintaxis que incluye un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado, en el que el primer elemento de sintaxis se codifica utilizando uno o más códigos de Golomb, de modo que la longitud del primer elemento de sintaxis codificado es menor o igual a un número máximo predeterminado de bits (902). El descodificador de vídeo 30 puede configurarse adicionalmente para descodificar la pluralidad de elementos de sintaxis, incluida la descodificación del primer elemento de sintaxis utilizando uno o más códigos de Golomb (904), reconstruir la paleta basándose en la pluralidad de elementos de sintaxis descodificados (906) y descodificar el bloque de datos de vídeo usando la paleta reconstruida (908). El descodificador de vídeo 30 también puede configurarse para mostrar el bloque descodificado de datos de vídeo.

35

40

45

[0213] En un ejemplo de la divulgación, el primer elemento de sintaxis es un elemento de sintaxis `palette_num_signalled_entries`. En otro ejemplo de la divulgación, la pluralidad de elementos de sintaxis incluye los valores de paleta indicados como señalizados explícitamente por el primer elemento de sintaxis.

50

[0214] En un ejemplo de la divulgación, el número máximo predeterminado de bits es 32, y el uno o más códigos de Golomb es un código exponencial Golomb de orden 0. En otro ejemplo de la divulgación, el número máximo predeterminado de bits es 32, y el uno o más códigos de Golomb es una concatenación de un código de Rice truncado y un código exponencial Golomb.

55

[0215] En otro ejemplo de la divulgación, se define un valor máximo del primer elemento de sintaxis en relación con un segundo elemento de sintaxis que indica un tamaño máximo de la paleta y un tercer elemento de sintaxis que indica un tamaño máximo de un predictor de paleta. En este ejemplo, el descodificador de vídeo 30 puede configurarse además para definir el segundo elemento de sintaxis como un valor de 0 a un primer umbral, y definir el tercer elemento de sintaxis como un valor de 0 a un segundo umbral. En un ejemplo, el primer umbral es uno de 4095 o 4096 y el segundo umbral es uno de 4095, 8191 u 8192.

60

[0216] En otro ejemplo de la divulgación, se define un valor máximo del primer elemento de sintaxis en relación con un segundo elemento de sintaxis que indica un tamaño máximo de la paleta y un tercer elemento de sintaxis que indica un tamaño máximo de un predictor de paleta. En este ejemplo, el descodificador de vídeo 30 puede configurarse además para definir que el segundo elemento de sintaxis sea menor o igual a un número de píxeles en el bloque más grande posible en el flujo de bits de vídeo codificado, y definir que el tercer elemento de sintaxis

65

sea menor o igual a $K \cdot$ un valor del segundo elemento de sintaxis, donde K es una constante positiva. En un ejemplo, K es 2.

5 **[0217]** En otro ejemplo de la divulgación, el descodificador de vídeo 30 puede configurarse además para recibir un elemento de sintaxis que indica un tipo de ejecución de paleta en el caso de que un píxel actual no sea un primer píxel en un orden de escaneo, e inferir el elemento de sintaxis que indica un tipo de ejecución de paleta en el caso de que el píxel actual sea el primer píxel en el orden de escaneo.

10 **[0218]** Si bien se han descrito anteriormente combinaciones particulares de diversos aspectos de las técnicas, estas combinaciones se proporcionan meramente para ilustrar ejemplos de las técnicas descritas en esta divulgación. En consecuencia, las técnicas de esta divulgación no deberían limitarse a estas combinaciones a modo de ejemplo y pueden abarcar cualquier combinación concebible de los diversos aspectos de las técnicas descritas en esta divulgación.

15 **[0219]** En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, 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 correspondan a un medio tangible tal como medios de
20 almacenamiento de datos, o medios de comunicación que incluyan cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo a 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 sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los
25 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 esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

30 **[0220]** A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor
35 u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente emiten datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de lo anterior también se deberían incluir dentro del alcance de los medios legibles
45 por ordenador.

[0221] 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 otros circuitos lógicos integrados o discretos equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o software dedicados, configurados para la codificación y la descodificación, o incorporados en un códec combinado.
50 Además, las técnicas se podrían implementar totalmente en uno o más circuitos o elementos lógicos.

[0222] Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar
60 aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec, o ser proporcionadas por un grupo de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

65

[0223] Se han descrito varios ejemplos de la divulgación. Se contempla cualquier combinación de los sistemas, operaciones o funciones descritos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de descodificación de datos de vídeo, comprendiendo el procedimiento:

5 recibir un bloque de datos de vídeo en un flujo de bits de vídeo codificado, habiendo sido codificado el bloque de datos de vídeo usando un modo de codificación basada en paleta;

10 recibir una pluralidad de elementos de sintaxis que son indicativos de una paleta que se usó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado, en el que un valor máximo del primer elemento de sintaxis se define en relación con uno o más de un segundo elemento de sintaxis que indica un tamaño máximo de la paleta o un tercer elemento de sintaxis que indica un tamaño máximo de un predictor de paleta, en el que el segundo elemento de sintaxis tiene un valor de 0 a un primer umbral y el tercer elemento de sintaxis tiene un valor de 0 a un segundo umbral, y se codifica utilizando uno o más códigos de Golomb, de manera que la longitud del primer elemento de sintaxis codificado es menor o igual que un número máximo predeterminado de bits;

20 descodificar la pluralidad de elementos de sintaxis, incluida la descodificación del primer elemento de sintaxis utilizando el uno o más códigos de Golomb;

reconstruir la paleta basándose en la pluralidad de elementos de sintaxis descodificados; y

25 descodificar el bloque de datos de vídeo usando la paleta reconstruida.

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

mostrar el bloque descodificado de datos de vídeo.

3. El procedimiento según la reivindicación 1, que comprende además:

30 recibir un elemento de sintaxis que indica un tipo de ejecución de paleta en el caso de que un píxel actual del bloque de datos de vídeo no sea un primer píxel en un orden de escaneo del bloque de datos de vídeo; y

35 inferir el elemento de sintaxis como indicador de un tipo de ejecución de paleta en el caso de que el píxel actual sea el primer píxel en el orden de escaneo.

4. Un aparato configurado para descodificar datos de vídeo, comprendiendo el aparato:

40 medios para recibir un bloque de datos de vídeo en un flujo de bits de vídeo codificado, habiendo sido codificado el bloque de datos de vídeo usando un modo de codificación basada en paleta;

45 medios para recibir una pluralidad de elementos de sintaxis que son indicativos de una paleta que se usó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en el flujo de bits de vídeo codificado, en el que un valor máximo del primer elemento de sintaxis se define en relación con uno o más de un segundo elemento de sintaxis que indica un tamaño máximo de la paleta o un tercer elemento de sintaxis que indica un tamaño máximo de un predictor de paleta, en el que el segundo elemento de sintaxis tiene un valor de 0 a un primer umbral y el tercer elemento de sintaxis tiene un valor de 0 a un segundo umbral, y se codifica utilizando uno o más códigos de Golomb de manera que la longitud del primer elemento de sintaxis codificado es menor que o igual a un número máximo predeterminado de bits;

50 medios para descodificar la pluralidad de elementos de sintaxis, incluida la descodificación del primer elemento de sintaxis utilizando el uno o más códigos de Golomb;

55 medios para reconstruir la paleta basándose en la pluralidad de elementos de sintaxis descodificados; y

60 medios para descodificar el bloque de datos de vídeo usando la paleta reconstruida.

5. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:

65 codificar un bloque de datos de vídeo usando una paleta y un modo de codificación basada en paleta;

generar una pluralidad de elementos de sintaxis que son indicativos de la paleta que se usó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se señalizan explícitamente en un flujo de bits de vídeo codificado;

5

codificar el primer elemento de sintaxis que se define en relación con uno o más de un segundo elemento de sintaxis que indica un tamaño máximo de la paleta o un tercer elemento de sintaxis que indica un tamaño máximo de un predictor de paleta, en el que el segundo elemento de sintaxis tiene un valor de 0 a un primer umbral y el tercer elemento de sintaxis tiene un valor de 0 a un segundo umbral, utilizando uno o más códigos de Golomb de manera que la longitud del primer elemento de sintaxis codificado sea menor o igual a un número máximo predeterminado de bits; e

10

incluir la pluralidad de elementos de sintaxis en el flujo de bits de vídeo codificado.

15 **6.** El procedimiento según la reivindicación 1 o 5, en el que el primer elemento de sintaxis es un elemento de sintaxis num_signalled_palette_entries.

7. El procedimiento según la reivindicación 1 o 5, en el que el número máximo predeterminado de bits es 32, y en el que el uno o más códigos de Golomb es un código exponencial Golomb de orden 0.

20

8. El procedimiento según la reivindicación 1 o 5, en el que el número máximo predeterminado de bits es 32, y en el que el uno o más códigos de Golomb es una concatenación de un código de Rice truncado y un código exponencial Golomb.

25 **9.** El procedimiento según la reivindicación 1 o 5, en el que la pluralidad de elementos de sintaxis incluye los valores de paleta indicados como señalizados explícitamente por el primer elemento de sintaxis.

10. El procedimiento según la reivindicación 1 o 5, en el que el primer umbral es uno de 4095 o 4096 y el segundo umbral es uno de 4095, 8191 u 8192.

30

11. El procedimiento según la reivindicación 1 o 5, en el que el valor máximo del primer elemento de sintaxis se define en relación con el segundo elemento de sintaxis, de la pluralidad de elementos de sintaxis, que indica un tamaño máximo de la paleta, y el tercer elemento de sintaxis, de la pluralidad de elementos de sintaxis, que indica el tamaño máximo del predictor de paleta, comprendiendo el procedimiento, además:

35

definir el segundo elemento de sintaxis para que sea menor que o igual a un número de píxeles en un bloque más grande posible de los datos de vídeo en el flujo de bits de vídeo codificado; y

definir el tercer elemento de sintaxis para que sea menor que o igual a $K * \text{valor del segundo elemento de sintaxis}$, donde K es una constante positiva y $*$ indica una operación de multiplicación.

40

12. El procedimiento según la reivindicación 1 o 5, en el que K es 2.

13. El procedimiento según la reivindicación 5, que comprende además:

45

señalizar un elemento de sintaxis que indica un tipo de ejecución de la paleta en el caso de que un píxel actual no sea un primer píxel en un orden de escaneo; y

no señalar el elemento de sintaxis que indica un tipo de ejecución de paleta en el caso de que el píxel actual del bloque de datos de vídeo sea el primer píxel en el orden de escaneo del bloque de datos de vídeo.

50

14. Un aparato configurado para codificar datos de vídeo, comprendiendo el aparato:

medios para codificar un bloque de datos de vídeo usando una paleta y un modo de codificación basada en paleta;

55

medios para generar una pluralidad de elementos de sintaxis que son indicativos de la paleta que se utilizó para codificar el bloque de datos de vídeo, incluyendo la pluralidad de elementos de sintaxis un primer elemento de sintaxis que indica un número de valores de paleta para la paleta que se indican explícitamente en un flujo de bits de vídeo codificado;

60

medios para codificar el primer elemento de sintaxis que se define en relación con uno o más de un segundo elemento de sintaxis que indica un tamaño máximo de la paleta, o un tercer elemento de sintaxis que indica un tamaño máximo de un predictor de paleta, en el que el segundo elemento de sintaxis tiene un valor de 0 a un primer umbral y el tercer elemento de sintaxis tiene un valor de 0 a un

65

segundo umbral, utilizando uno o más códigos de Golomb de manera que la longitud del primer elemento de sintaxis codificado sea menor que o igual a un número máximo predeterminado de bits; y

medios para incluir la pluralidad de elementos de sintaxis en el flujo de bits de vídeo codificado.

5

15. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo configurado para codificar datos de vídeo lleven a cabo el procedimiento según cualquiera de las reivindicaciones 1 a 3 o 5 a 13.

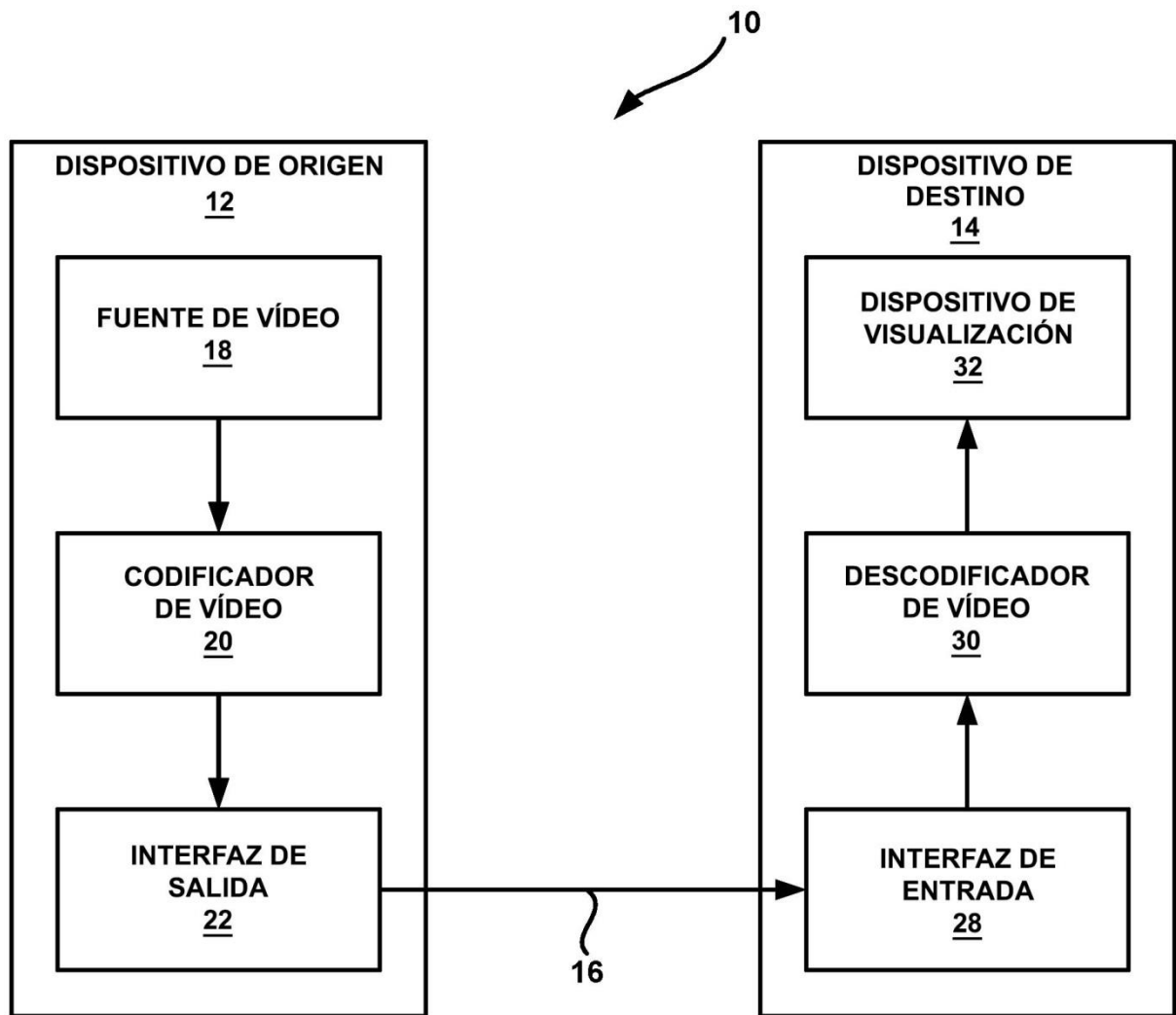


FIG. 1

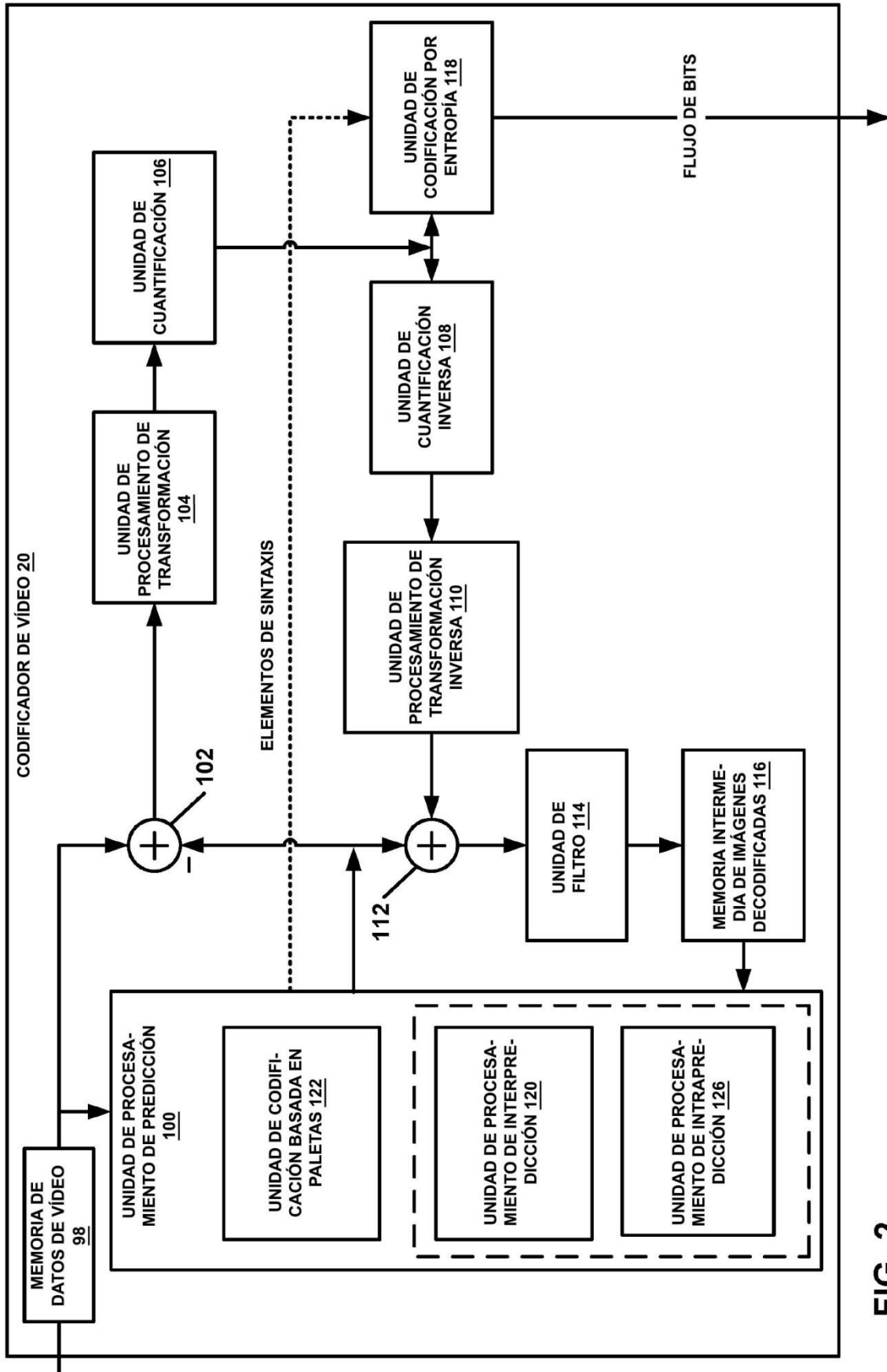


FIG. 2

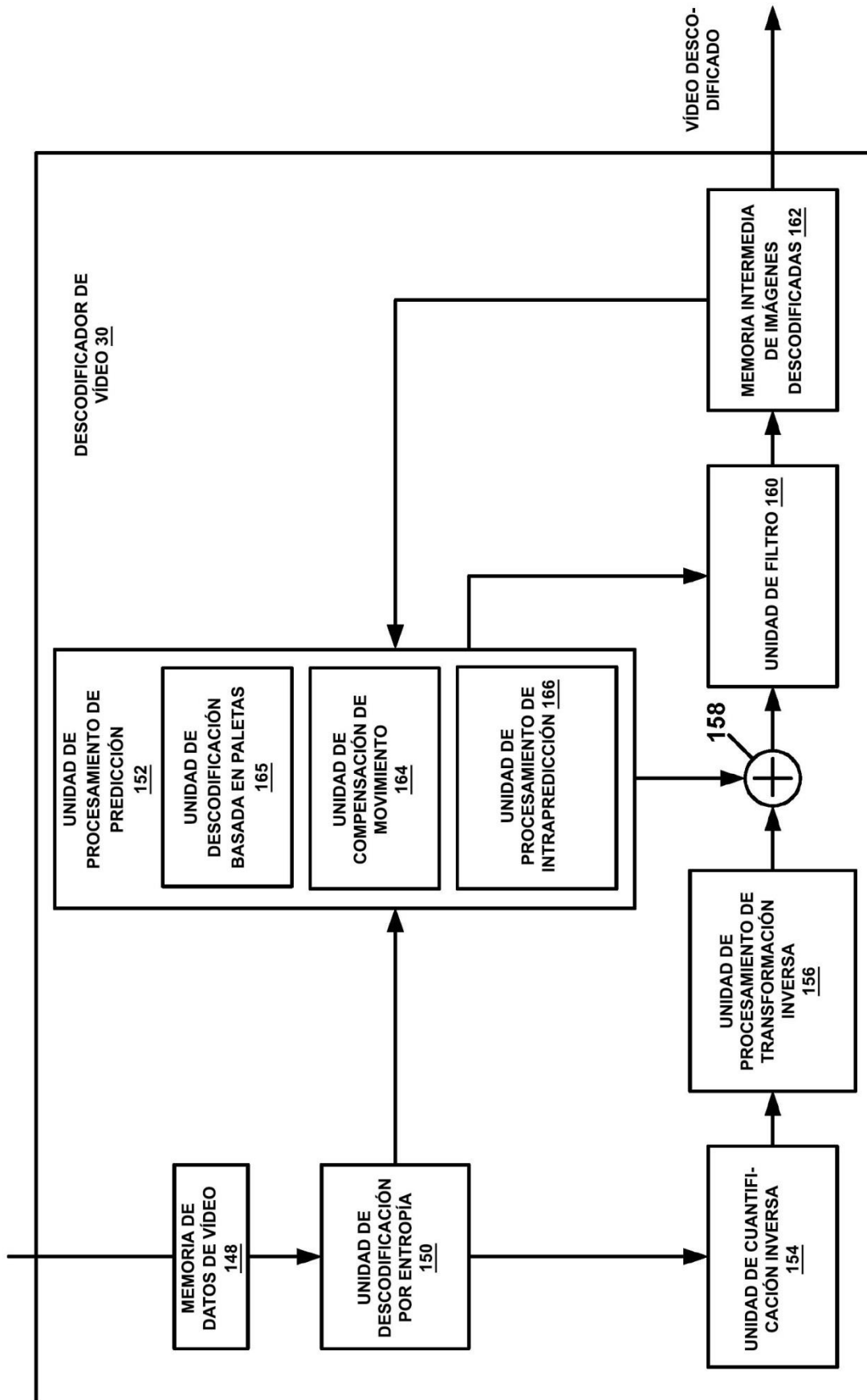


FIG. 3

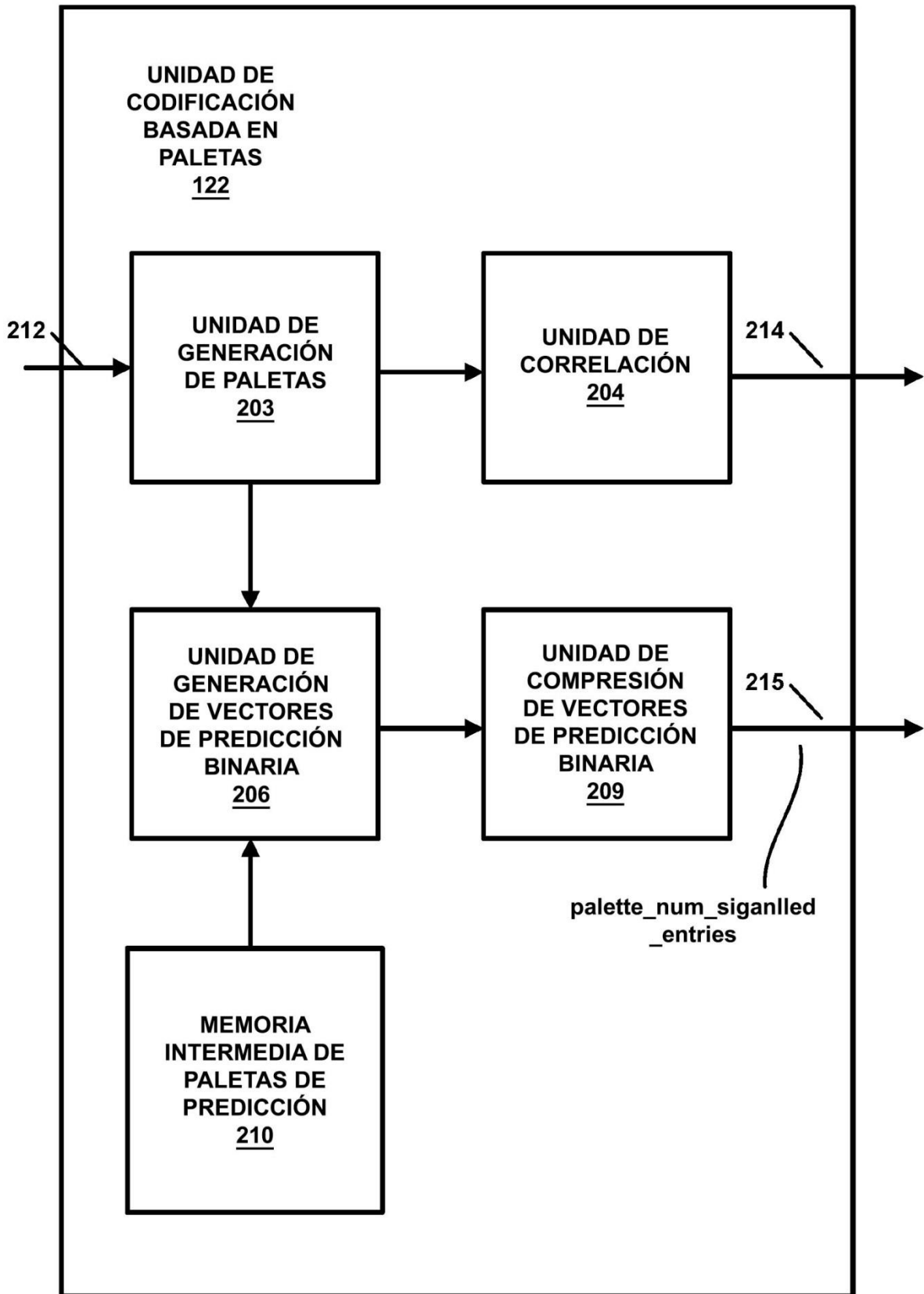


FIG. 4

**MEMORIA INTERMEDIA
DE PALETAS DE
PREDICCIÓN 210**

ÍNDICE DE ENTRADA	VALOR DE PÍXEL
1	VALOR A
2	VALOR B
3	VALOR C
4	VALOR D
5	VALOR E
6	VALOR F
7	VALOR G
8	VALOR H
9	VALOR I
10	VALOR J
11	VALOR K
12	VALOR L
13	VALOR M
...	...

PALETA ACTUAL 220

ÍNDICE DE ENTRADA	VALOR DE PÍXEL
1	VALOR A
2	VALOR B
3	VALOR E
4	VALOR I
5	NUEVO VALOR
6	NUEVO VALOR
7	NUEVO VALOR
8	NUEVO VALOR

b = [1100100010000]

FIG. 5

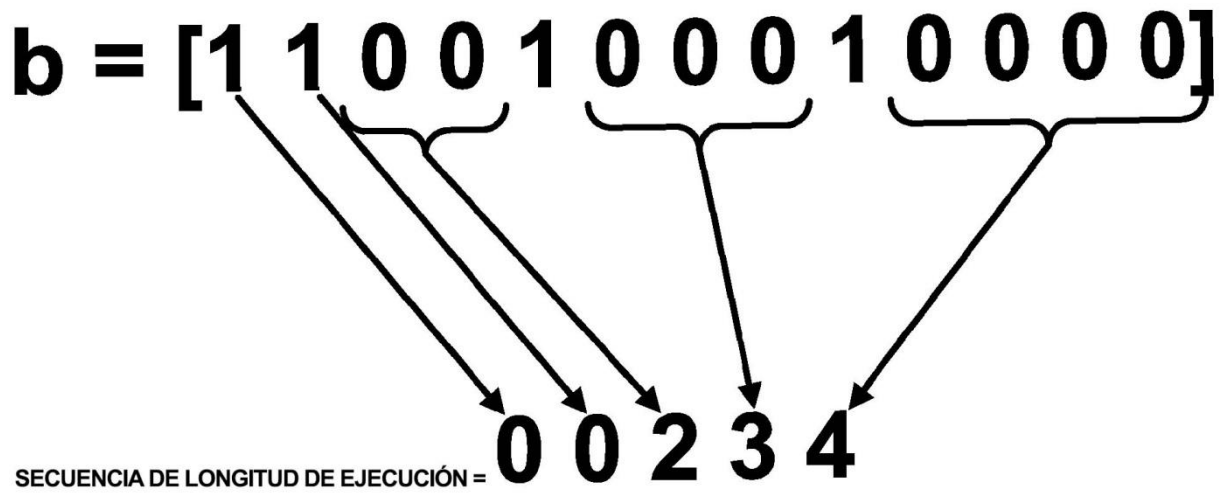


FIG. 6

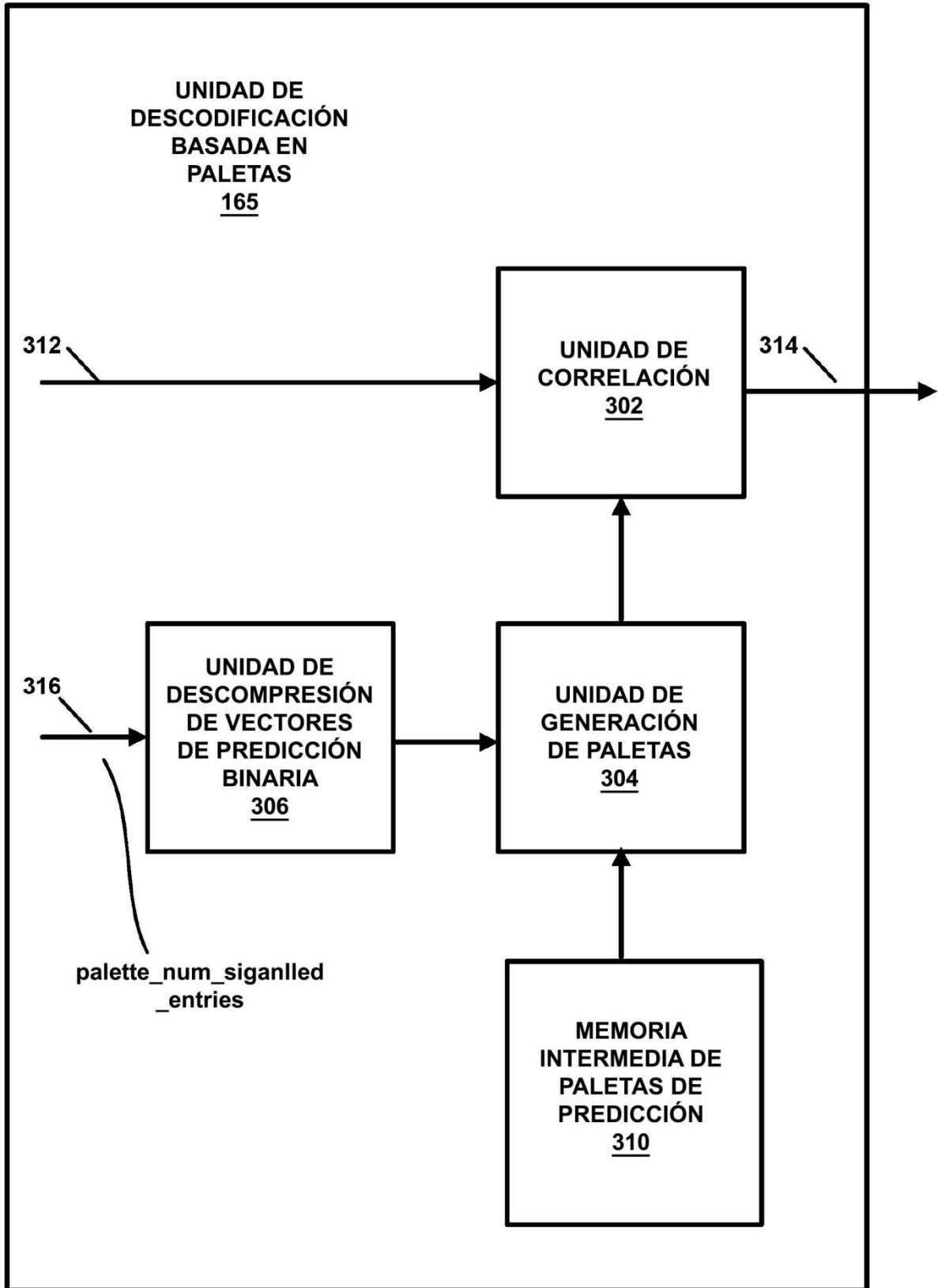


FIG. 7

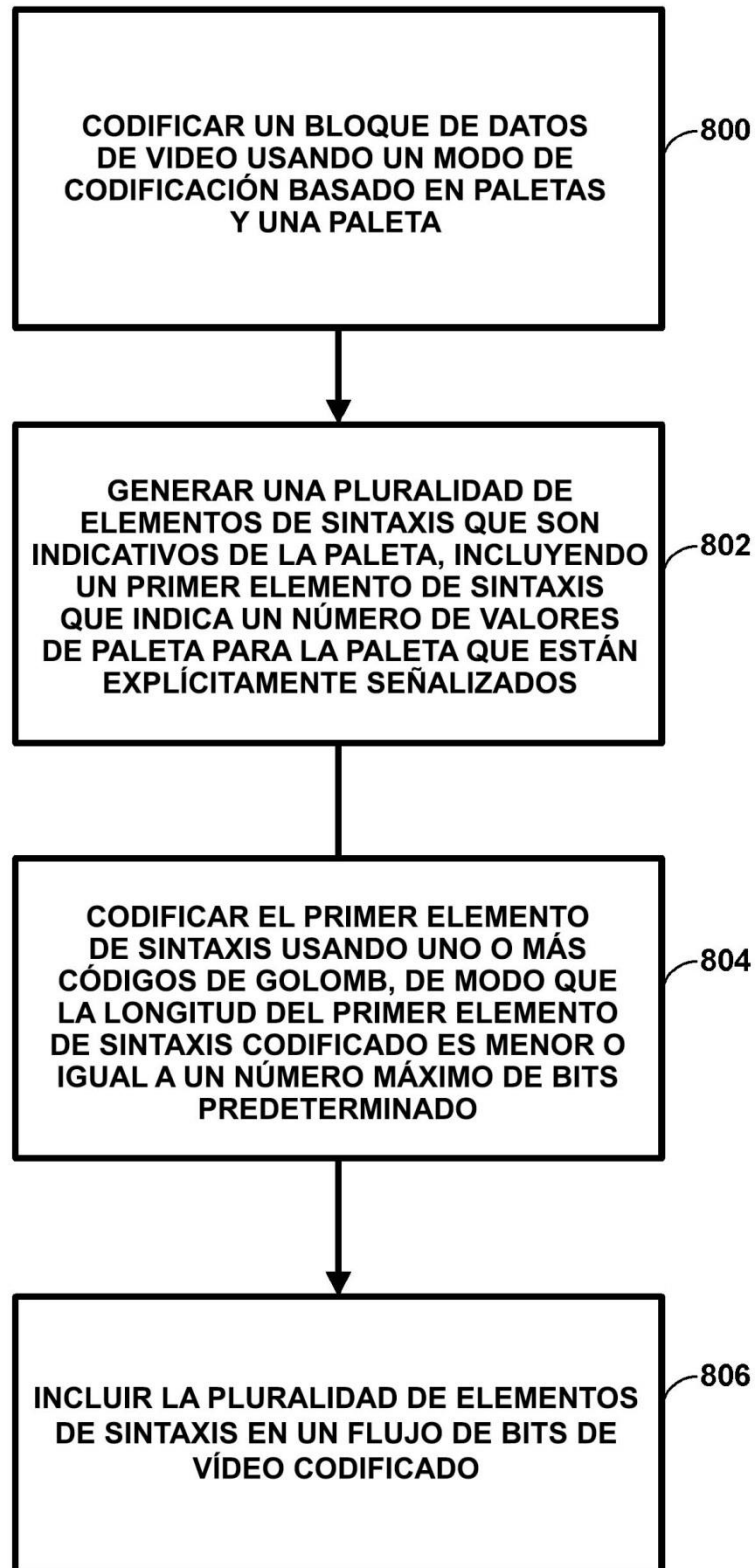


FIG. 8

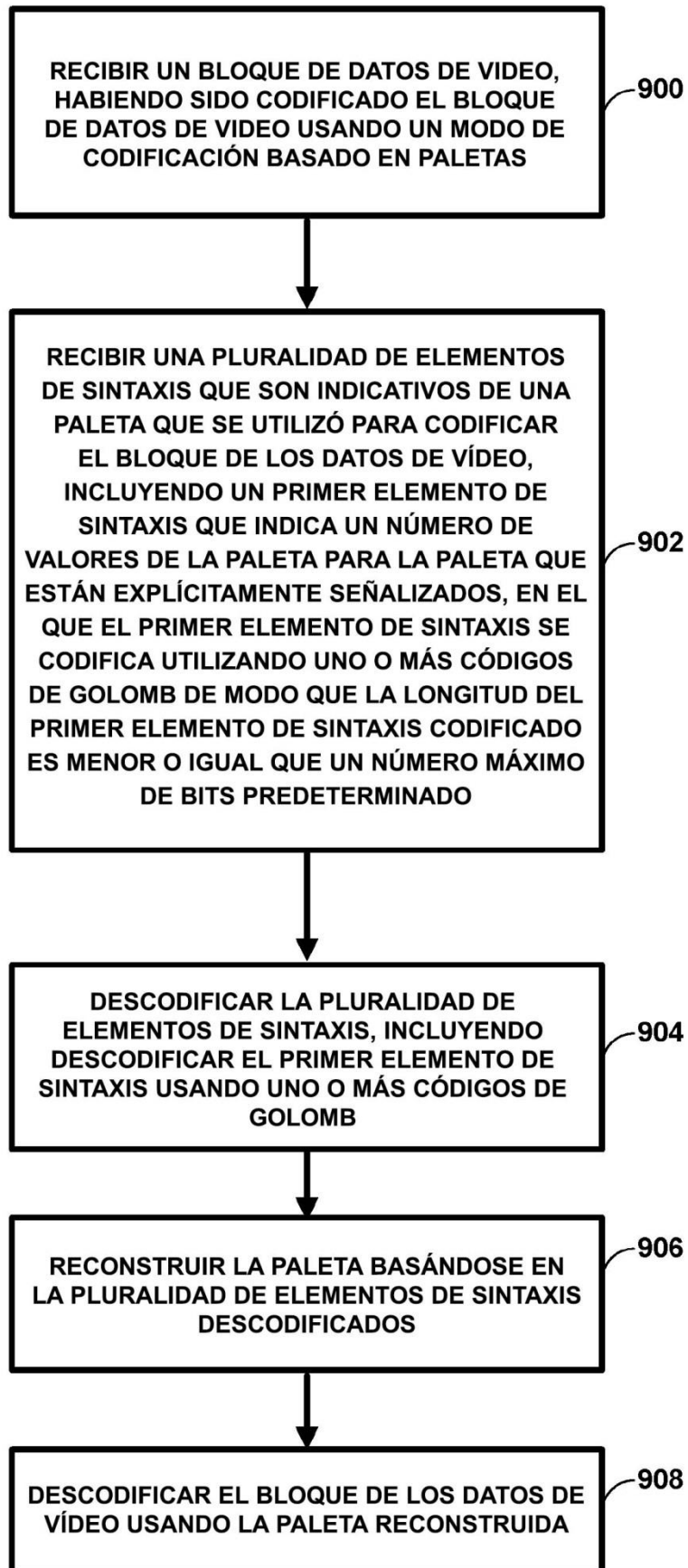


FIG. 9