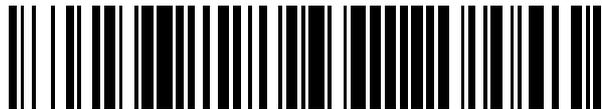


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 750 957**

51 Int. Cl.:

H04N 19/27 (2014.01)

H04N 19/93 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **25.03.2015 PCT/US2015/022468**

87 Fecha y número de publicación internacional: **01.10.2015 WO15148652**

96 Fecha de presentación y número de la solicitud europea: **25.03.2015 E 15715588 (8)**

97 Fecha y número de publicación de la concesión europea: **24.07.2019 EP 3123719**

54 Título: **Señalización de predictores de paleta con código de longitud de ejecución para codificación de vídeo**

30 Prioridad:

25.03.2014 US 201461970257 P
17.04.2014 US 201461981105 P
23.05.2014 US 201462002668 P
20.06.2014 US 201462015327 P
27.06.2014 US 201462018461 P
24.08.2014 US 201462041119 P
24.03.2015 US 201514667411

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
30.03.2020

73 Titular/es:

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

72 Inventor/es:

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

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 750 957 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

Señalización de predictores de paleta con código de longitud de ejecución para codificación de vídeo

5 **CAMPO TÉCNICO**

[0001] Esta divulgación se refiere a la codificación y a la decodificación de vídeo.

ANTECEDENTES

10

[0002] Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tablets electrónicas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión continua 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 eficacia (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 más eficazmente implementando dichas técnicas de compresión de vídeo.

15

20

25

30

[0003] Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) 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) se puede dividir en bloques de vídeo. Los bloques de vídeo de un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a unas muestras de referencia de bloques contiguos de la misma imagen. Los bloques de vídeo de un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a unas muestras de referencia de bloques contiguos de la misma imagen, o predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.

35

40

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va 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 un modo de codificación intra y los datos residuales. Para una mayor compresión, los datos residuales se pueden transformar desde el dominio del píxel hasta un dominio de transformada, dando como resultado coeficientes residuales, que a continuación se pueden cuantificar. Los coeficientes cuantificados, dispuestos inicialmente en una matriz bidimensional, se pueden escanear a fin de generar un vector unidimensional de coeficientes, y se puede aplicar codificación por entropía para lograr incluso más compresión.

45

[0005] Chen *et al.*, en "Description of screed content coding technology proposal by Qualcomm" (JCTVC-Q0031) divulgan la señalización de la paleta en la sección 2.7.1.1.

[0006] El documento US 2012/0057799 divulga dos formas de codificación de longitud de ejecución.

SUMARIO

50

55

60

65

[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. A continuación, el decodificador de vídeo puede usar el índice en la paleta para determinar qué valor de color usar para un píxel particular en el bloque.

[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 previamente codificados. Esta divulgación describe técnicas para señalar uno o más elementos sintácticos (por ejemplo, un vector de predicción binaria) que indican qué entradas de paleta de bloques de datos de vídeo previamente codificados pueden reutilizarse para un

bloque actual de datos de vídeo. Esta divulgación describe además técnicas para codificar y descodificar el vector de predicción binaria.

5 [0009] Se divulga un procedimiento para descodificar datos de vídeo, un procedimiento para codificar datos de vídeo, un aparato configurado para descodificar datos de vídeo y un aparato configurado para codificar datos de vídeo, como se define en las reivindicaciones independientes. También se divulga un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores lleven a cabo un procedimiento definido en las reivindicaciones 1-7.

10 [0010] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

15 BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0011]

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

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

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

La FIG. 4 es un diagrama de bloques que ilustra una unidad de codificación basada en paleta a modo de ejemplo del codificador de vídeo de la FIG. 2.

30 La FIG. 5 es un diagrama conceptual que ilustra una técnica de predicción de paleta a modo de ejemplo de acuerdo con las técnicas de la divulgación.

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

La FIG. 7 es un diagrama de bloques que ilustra una unidad de descodificación basada en paleta a modo de ejemplo del codificador de vídeo de la FIG. 3.

40 La FIG. 8 es un diagrama de flujo que ilustra un procedimiento de codificación de ejemplo de la divulgación.

La FIG. 9 es un diagrama de flujo que ilustra un procedimiento de descodificación de ejemplo de la divulgación.

45 La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de determinación de entradas de paleta para la codificación de vídeo basada en paleta, congruente con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

50 [0012] Esta divulgación incluye técnicas para codificación y compresión de vídeo. En particular, esta divulgación describe técnicas para codificación basada en paleta de datos de vídeo. En la codificación de vídeo tradicional, se supone que las imágenes son de tono continuo y espacialmente homogéneas. 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.

55 [0013] Sin embargo, en aplicaciones como escritorio remoto, trabajo colaborativo y pantalla inalámbrica, el contenido de pantalla generado por ordenador (por ejemplo, tal como texto o gráficos de ordenador) puede ser el contenido dominante a comprimir. Este tipo de contenido tiende a tener tonos discretos y líneas características nítidas y límites de objetos de alto contraste. Es posible que la suposición de tono continuo y suavidad ya no se aplique al contenido de pantalla, y por tanto que las técnicas tradicionales de codificación de vídeo puedan no ser formas eficientes de comprimir los datos de vídeo, incluido el contenido de pantalla.

60 [0014] Las técnicas de codificación basada en paleta pueden ser particularmente adecuadas para la codificación de contenido generado por la 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 un descodificador de vídeo) puede formar una denominada "paleta" como una tabla de colores para representar los datos de vídeo del área particular (por ejemplo, un bloque dado). Cada píxel se puede asociar con una entrada de

65

la paleta que representa el color del píxel. Por ejemplo, el codificador de vídeo puede codificar un índice que relaciona el valor de píxel con el valor adecuado en la paleta.

5 **[0015]** Las técnicas para codificación basada en paleta de datos de vídeo se pueden usar con una o más técnicas de codificación diferentes, tales como técnicas para codificación interpredictiva o intrapredictiva. Por ejemplo, como se describe con más detalle a continuación, un codificador o descodificador, o un codificador-descodificador combinado (códec), puede estar configurado para realizar una codificación interpredictiva e intrapredictiva, así como una codificación basada en paleta.

10 **[0016]** En algunos ejemplos, las técnicas de codificación basada en paleta pueden estar configuradas para su uso con uno o más estándares 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.

20 **[0017]** 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, 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", 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>.

30 **[0018]** Al utilizar técnicas de codificación basadas en paleta de ejemplo, un codificador de vídeo puede codificar un bloque de datos de vídeo determinando una paleta para el bloque (por ejemplo, codificando la paleta explícitamente, prediciéndola o combinando ambos casos), ubicando una entrada en la paleta para representar el valor de cada píxel y codificando el bloque con valores de índices para los píxeles que relacionan el valor de píxel con la paleta. Un descodificador de vídeo puede obtener, a partir de una corriente de bits codificada, una paleta para un bloque, así como valores de índices para los píxeles del bloque. El descodificador de vídeo puede relacionar los valores de índice de los píxeles con las entradas de la paleta para reconstruir los valores de píxel del bloque. El ejemplo anterior pretende proporcionar una descripción general de la codificación basada en paleta.

40 **[0019]** Además de señalar los valores de índice 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 previamente codificados. Las técnicas de predicción de paleta pueden incluir la señalización de una secuencia de elementos sintácticos (por ejemplo, un vector de predicción binaria) donde el valor de cada elemento sintáctico en el vector de predicción binaria indica si una entrada de paleta de un bloque de datos de vídeo previamente codificado puede reutilizarse para un bloque actual de datos de vídeo. Tales vectores de predicción binaria pueden ser largos para tamaños de paleta grandes y/o para casos en los que se almacena una gran cantidad de entradas de paleta para su posible uso en el proceso de predicción de paleta. Los vectores de predicción binaria grandes pueden dar como resultado niveles de velocidad de bits inaceptables. En vista de estos inconvenientes, esta divulgación describe técnicas para codificar y descodificar vectores de predicción binaria en un proceso de predicción de paleta.

50 **[0020]** La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación de vídeo 10 que puede utilizar las técnicas de esta divulgación. Tal como se utiliza en el presente documento, el término "codificador de vídeo" se refiere genéricamente tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" pueden referirse genéricamente a la codificación de vídeo o la descodificación de vídeo. El codificador de vídeo 20 y el descodificador de vídeo 30 del sistema de codificación de vídeo 10 representan ejemplos de dispositivos que pueden estar configurados para realizar técnicas para 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 de Vídeo de Alta Eficiencia (HEVC), usando la codificación basada en paleta o bien la codificación no basada en paleta. Los modos de codificación no basados en paleta pueden referirse a diversos modos de codificación temporal interpredictiva o a modos de codificación espacial intrapredictiva, tales como los diversos modos de codificación especificados por la versión 1 de la HEVC.

[0021] Como se muestra en la FIG. 1, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 y un dispositivo de destino 14. El dispositivo de origen 12 genera datos de vídeo codificados. En consecuencia, el dispositivo de origen 12 se puede denominar dispositivo de codificación de vídeo o aparato de codificación de vídeo. El dispositivo de destino 14 puede descodificar los datos de vídeo codificados generados por el dispositivo de origen 12. En consecuencia, el dispositivo de destino 14 se puede denominar dispositivo de descodificación de vídeo o aparato de descodificación de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 pueden ser ejemplos de dispositivos de codificación de vídeo o aparatos de codificación de vídeo. Diversas implementaciones del dispositivo de origen 12, el dispositivo de destino 14, o ambos, pueden incluir uno o más procesadores y memoria acoplada a uno o más procesadores. La memoria puede incluir, pero no se limita a RAM, ROM, EEPROM, memoria flash o cualquier otro medio que pueda usarse para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda acceder un ordenador, como se describe en el presente documento.

[0022] El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia gama de dispositivos, que incluye ordenadores de escritorio, dispositivos informáticos móviles, ordenadores tipo notebook (por ejemplo, portátiles), ordenadores tipo tablet, descodificadores multimedia, aparatos telefónicos tales como los denominados teléfonos "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores para vehículos o similares.

[0023] El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 por medio de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicación que permiten al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de origen 12 puede modular los datos de vídeo codificados de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. El uno o más medios de comunicación pueden incluir medios de comunicación inalámbrica y/o alámbrica, tales como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El uno o más medios de comunicación pueden formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global (por ejemplo, Internet). El uno o más medios de comunicación pueden incluir encaminadores, conmutadores, estaciones base u otros equipos que facilitan la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

[0024] En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacena datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento, por ejemplo, mediante el acceso a disco o el acceso a tarjeta. El medio de almacenamiento puede incluir una variedad de medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria *flash* u otros medios de almacenamiento digital adecuados para almacenar datos de vídeo codificados.

[0025] En otro ejemplo, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacena los datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder a datos de vídeo codificados almacenados en el servidor de archivos o en otro dispositivo de almacenamiento intermedio mediante 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 a un dispositivo de destino 14. Entre los ejemplos de servidores de archivos se incluyen servidores web (por ejemplo, para una página web), servidores de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados en red (NAS), servidores de transmisión continua HTTP y unidades de disco local.

[0026] 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. Los ejemplos de tipos de conexiones de datos pueden incluir canales inalámbricos (por ejemplo, conexiones wifi), conexiones alámbricas (por ejemplo, DSL, módem de cable, etc.) o combinaciones de ambas que sean adecuadas para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0027] Las técnicas de esta divulgación para un modo de codificación basada en paleta en codificación de vídeo no están limitadas a aplicaciones o a configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como soporte de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, por medio de Internet, codificación de datos de vídeo para su almacenamiento en un medio de almacenamiento de datos, descodificación de datos de vídeo almacenados en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 puede estar configurado para admitir transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

[0028] El sistema de codificación de vídeo 10 ilustrado en la FIG. 1 es simplemente un ejemplo y las técnicas de esta divulgación pueden aplicarse a configuraciones de codificación de vídeo (por ejemplo, codificación de vídeo o descodificación de vídeo) que no incluyan necesariamente ninguna comunicación de datos entre los dispositivos de codificación y descodificación. En otros ejemplos, los datos se recuperan desde una memoria local, se transmiten en continuo por una red o similares. Un dispositivo de codificación de vídeo puede codificar y almacenar datos en una memoria, y/o un dispositivo de descodificación de vídeo puede recuperar y descodificar datos de una memoria. En muchos ejemplos, la codificación y la descodificación se realizan mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en una memoria y/o recuperan y descodifican datos de una memoria.

[0029] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos ejemplos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. La fuente de vídeo 18 puede incluir un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contenga datos de vídeo previamente capturados, una interfaz de alimentación de vídeo para recibir datos de vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de vídeo, o una combinación de dichas fuentes de datos de vídeo.

[0030] 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.

[0031] En el ejemplo de la FIG. 1, el dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos ejemplos, la interfaz de entrada 28 incluye un receptor y/o un módem. La interfaz de entrada 28 puede recibir 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.

[0032] 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. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar incluidos cada uno en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado formando parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

[0033] 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 "señalar" o "transmitir" puede referirse en general a la comunicación de elementos sintácticos 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 sintácticos 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.

[0034] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 funcionan de acuerdo con una norma de compresión de vídeo, tal como la norma HEVC mencionada anteriormente y descrita en la versión 1 de la HEVC. 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 modos 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 paleta se pueden aplicar a codificadores y descodificadores configurados para funcionar de acuerdo con otras normas o futuras normas de codificación de vídeo. Por consiguiente, la aplicación de un modo de codificación basada en paleta 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.

[0035] En varios ejemplos, las técnicas descritas en esta divulgación pueden incluir técnicas para diversas combinaciones de paletas de determinación, predicción y/o señalización en la codificación basada en paleta. Como se describe con mayor detalle a continuación, las técnicas pueden incluir cualquier combinación de determinar predictores de paleta, generar un vector de predicción binaria y codificar/descodificar el vector de predicción binaria. Si bien ciertos ejemplos pueden describirse individualmente con fines ilustrativos y de claridad, esta divulgación contempla cualquier combinación de las técnicas para la codificación basada en paleta descritas en el presente documento.

[0036] Como se explicará con más detalle a continuación con referencia a las FIGS. 4-9, el codificador de vídeo 20 puede configurarse para generar una paleta para un bloque actual de los datos de vídeo, generar un vector de predicción binaria para la paleta para el bloque actual de datos de vídeo, con el vector de predicción binaria que comprende entradas que indican si entradas de paleta usadas previamente (es decir, entradas en paletas para bloques previamente codificados) se reutilizan para la paleta para el bloque actual de datos de vídeo, codificar el vector de predicción binaria usando una técnica de codificación de longitud de ejecución y codificar el bloque actual de datos de vídeo usando la paleta. Del mismo modo, el descodificador de vídeo 30 puede configurarse para recibir un vector de predicción binaria codificada para un bloque actual de los datos de vídeo, descodificar el vector de predicción binaria codificada utilizando una técnica de descodificación de longitud de ejecución, generar una paleta para el bloque actual de datos de vídeo basado en el vector de predicción binaria, con el vector de predicción binaria que comprende entradas que indican si las entradas de paleta usadas anteriormente se reutilizan o no para la paleta para el bloque actual de datos de vídeo, y descodificar el bloque actual de datos de vídeo usando la paleta.

[0037] 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 con S_L , S_{Cb} y S_{Cr} . S_L es una matriz bidimensional (es decir, un bloque) de muestras de luma. S_{Cb} es una matriz bidimensional de muestras de crominancia Cb. S_{Cr} es una matriz bidimensional de muestras de crominancia Cr. Las muestras de crominancia también se pueden denominar en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede incluir solo una matriz de muestras de luma.

[0038] Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede ser un bloque de árbol de codificación de muestras de luma, dos 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 $N \times N$ de muestras. Una CTU también puede denominarse "bloque de árbol" o "unidad de codificación de máximo tamaño" (LCU). Las CTU de HEVC pueden ser análogas en términos generales a los macrobloques de otros estándares, tales como H.264/AVC. Sin embargo, una CTU no está necesariamente limitada 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.

[0039] 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 aplica 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 bloques de predicción. El codificador de vídeo 20 puede generar bloques de predicción de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU.

[0040] El codificador de vídeo 20 puede usar predicción intra o predicción inter para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 usa predicción intra para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU basándose en unas muestras descodificadas de la imagen asociada a la PU.

[0041] Si el codificador de vídeo 20 usa predicción inter para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU basándose en unas 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.

[0042] Después de que el codificador de vídeo 20 genera uno o más 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 un bloque residual para la CU. Cada muestra en el bloque residual puede indicar una diferencia entre una muestra en uno de los bloques predictivos de la CU y una muestra correspondiente en un bloque de codificación de la CU. Por ejemplo, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra en el bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques de predicción de luma de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual de Cb para la CU. Cada muestra en el bloque residual de Cb de la CU puede indicar una diferencia entre una muestra Cb en uno de los bloques Cb predictivos de la CU y una muestra correspondiente en el bloque de codificación 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.

[0043] Además, el codificador de vídeo 20 puede usar la partición en árbol cuaternario para descomponer los bloques residuales (por ejemplo, los bloques residuales de luma, Cb y Cr) de una CU en uno o más bloques de transformada (por ejemplo, bloques de transformada de luma, Cb y Cr). Un bloque de transformada puede ser un bloque rectangular de muestras en el que se aplique la misma transformada. Una unidad de transformada (TU) de una CU puede ser un bloque de transformada de muestras de luma, dos bloques de transformada correspondientes de muestras de croma y estructuras sintácticas usadas para transformar las muestras de bloques de transformada. Por tanto, cada TU de una CU puede estar asociada a un bloque de transformada de luma, un bloque de transformada de Cb y un bloque de transformada de Cr. El bloque de transformada de luma asociado con la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformada de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un subbloque del bloque residual de Cr de la CU.

[0044] El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de una TU para generar un bloque de coeficientes para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. Por ejemplo, el codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU para generar un bloque de coeficientes de luma para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cb de una TU para generar un bloque de coeficientes de Cb para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

[0045] 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 unos coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos sintácticos que indiquen los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar una codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos sintácticos que indican los coeficientes de transformada cuantificados. El codificador de vídeo 20 puede enviar los elementos sintácticos codificados por entropía en un flujo de bits.

[0046] Para aplicar la codificación CABAC a un elemento sintáctico, el codificador de vídeo puede binarizar el elemento sintáctico para formar una serie de uno o más bits, que se denominan "bins". Además, el codificador de vídeo puede identificar un contexto de codificación. El contexto de codificación puede identificar probabilidades de codificación de bins que tengan valores particulares. Por ejemplo, un contexto de codificación puede indicar una probabilidad de 0,7 de codificar un bin de valor 0 y una probabilidad de 0,3 de codificar un bin de valor 1. Después de identificar el contexto de codificación, el codificador de vídeo 20 puede dividir un intervalo en un subintervalo inferior y en un subintervalo superior. Uno de los subintervalos puede estar asociado con el valor 0 y el otro subintervalo puede estar asociado con el valor 1. El ancho de los subintervalos puede ser proporcional a las probabilidades indicadas para los valores asociados por el contexto de codificación identificado. Si un bin del elemento sintáctico tiene el valor asociado con el subintervalo inferior, el valor codificado puede ser igual al límite inferior del subintervalo inferior. Si el mismo bin del elemento sintáctico tiene el valor asociado con el subintervalo superior, el valor codificado puede ser igual al límite inferior del subintervalo superior. Para codificar el siguiente bin del elemento sintáctico, el codificador de vídeo 20 puede repetir estos pasos con el intervalo estando el subintervalo asociado con el valor del bit codificado. Cuando el codificador de vídeo 20 repite estos pasos para el siguiente bin, el codificador de vídeo puede usar probabilidades modificadas basadas en las probabilidades indicadas por el contexto de codificación identificado y en los valores reales de los bins codificados.

[0047] Cuando un descodificador de vídeo 30 realice la descodificación CABAC en un elemento sintáctico, el descodificador de vídeo puede identificar un contexto de codificación... El descodificador de vídeo puede a

continuación dividir un intervalo en un subintervalo inferior y un subintervalo superior. Uno de los subintervalos puede estar asociado con el valor 0 y el otro subintervalo puede estar asociado con el valor 1. El ancho de los subintervalos puede ser proporcional a las probabilidades indicadas para los valores asociados por el contexto de codificación identificado. Si el valor codificado está dentro del subintervalo más bajo, el descodificador de vídeo puede descodificar un bin que tenga el valor asociado con el subintervalo inferior. Si el valor codificado está dentro del subintervalo superior, el descodificador de vídeo puede descodificar un bin que tenga el valor asociado con el subintervalo superior. Para descodificar un siguiente bin del elemento sintáctico, el descodificador de vídeo puede repetir estos pasos siendo el intervalo el subintervalo que contenga el valor codificado. Cuando el descodificador de vídeo 30 repita estos pasos para el siguiente bin, el descodificador de vídeo 30 puede usar probabilidades modificadas basadas en las probabilidades indicadas por el contexto de codificación identificado y en los bins descodificados. A continuación, el descodificador de vídeo puede desbinarizar los bins para recuperar el elemento sintáctico.

[0048] En lugar de llevar a cabo una codificación CABAC convencional en todos los elementos sintácticos, el codificador de vídeo puede codificar algunos elementos sintácticos (por ejemplo, bins) usando codificación CABAC por derivación. Puede ser computacionalmente menos costoso realizar una codificación CABAC de derivación en un bin que realizar una codificación CABAC regular en el bin. Además, llevar a cabo la codificación CABAC de derivación puede permitir un mayor grado de paralelización y rendimiento. Los bins codificados usando la codificación CABAC de derivación pueden denominarse "bins de derivación". Agrupar bins de derivación juntos puede aumentar el rendimiento del codificador y descodificador de vídeo. El motor de codificación CABAC de derivación puede codificar varios bins en un solo ciclo, mientras que el motor de codificación CABAC normal puede codificar solo un único bin en un ciclo. El motor de codificación CABAC de derivación puede ser más simple porque el motor de codificación CABAC de derivación no selecciona contextos y puede asumir una probabilidad de $\frac{1}{2}$ para ambos símbolos (0 y 1). En consecuencia, al obtener la codificación CABAC, los intervalos se dividen directamente a la mitad.

[0049] El codificador de vídeo 20 puede enviar un flujo de bits que incluya los elementos sintácticos codificados por entropía. El flujo de bits puede incluir una secuencia de bits que forme una representación de imágenes codificadas y datos asociados. En otras palabras, el flujo de bits puede comprender una representación codificada de datos de vídeo. 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 unidad NAL puede incluir un elemento sintáctico que indica un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por la cabecera de unidad NAL de una unidad NAL indica el tipo de la unidad NAL. Una RBSP puede ser una estructura sintáctica que contiene un número entero de bytes que están encapsulados dentro de una unidad NAL. En algunos casos, una RBSP incluye bits cero.

[0050] Diferentes tipos de unidades NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad de NAL puede encapsular una RBSP para un conjunto de parámetros de imagen (PPS), un segundo tipo de unidad de NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad de NAL puede encapsular una RBSP para SEI, etc. 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).

[0051] El descodificador de vídeo 30 puede recibir una corriente de bits generada por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede analizar el flujo de bits para descodificar elementos sintácticos 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 sintácticos descodificados a partir del flujo de bits. El proceso para reconstruir los datos de vídeo puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20. Por ejemplo, el descodificador de vídeo 30 puede usar los MV de las PU para determinar bloques predictivos para las PU de una CU actual. Además, el descodificador de vídeo 30 puede cuantificar inversamente bloques de coeficientes de transformada asociados a las TU de la CU actual. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes de transformada para reconstruir los bloques de transformada asociados a las TU de la CU actual. El descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para unas PU de la CU actual a unas muestras correspondientes de los bloques de transformada de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

[0052] Como se describió anteriormente, las técnicas descritas en el presente documento pueden aplicarse a la codificación basada en paleta de datos de vídeo. La codificación basada en paleta en general puede incluir una o más de las siguientes técnicas. Con la codificación de vídeo basada en paleta, un codificador de vídeo (por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30) puede formar una denominada "paleta" como una tabla de colores para representar los datos de vídeo del área particular (por ejemplo, un bloque dado). Cada píxel se puede asociar con una entrada de la paleta que representa el color del píxel. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar un índice que relaciona el valor de píxel con el valor adecuado de la paleta.

[0053] Con respecto al marco de la HEVC, como ejemplo, las técnicas de codificación basada en paleta pueden estar configuradas para su uso como un modo de unidad de codificación (CU). En otros ejemplos, las técnicas de codificación basada en paleta pueden estar configuradas para su uso como un modo PU en el marco de la HEVC. En consecuencia, todos los procesos siguientes divulgados, descritos en el contexto de un modo CU, se pueden aplicar, de forma adicional o alternativa, a las PU. Sin embargo, estos ejemplos basados en HEVC no se deberían considerar como una restricción o limitación de las técnicas de codificación basada en paleta, descritas en el presente documento, ya que dichas técnicas se pueden aplicar para que funcionen independientemente o como parte de otros sistemas/estándares existentes o todavía por elaborar. En estos casos, la unidad para la codificación por paleta pueden ser bloques cuadrados, bloques rectangulares o incluso regiones de forma no rectangular.

[0054] En el documento X. Guo y A. Saxena, "RCE4: Summary report of HEVC Range Extension Core Experiments 4 (RCE4) on palette coding for screen content", JCTVC-P0035, San José, Estados Unidos, 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.

[0055] En un procedimiento de ejemplo, como se describe, por ejemplo, en el documento X. Guo. Y. Lu, y S. Li, "RCE4: Test 1. Major-color-based screen content coding", JCTVC-P0108, San José, Estados Unidos, 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.

[0056] Al utilizar la clasificación, los píxeles de una unidad de codificación (CU) se pueden convertir en índices de color. Después de eso, los valores y el número de color principal se codifican. A continuación, 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, toda la línea 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.

[0057] En otro procedimiento de ejemplo, como se describe, por ejemplo, en el documento 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", JCTVC-P0198, San José, Estados Unidos, 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 contiguos superiores, respectivamente.
 - "Modo píxel": Primero se transmite un indicador de predicción. El valor de 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.

[0058] 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 previamente codificada (por ejemplo, como se indicó anteriormente con respecto a la "transmisión de la paleta").

[0059] En algunos ejemplos, el codificador de vídeo puede generar una lista de predictores de paleta cuando predice entradas de paleta. Por ejemplo, el documento C. Gisquet, G. Laroche y P. Onno, "AhG10: Palette predictor stuffing", 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.

[0060] En algunos ejemplos, todos los elementos de la lista de predictores de paleta se obtienen de la paleta previamente codificada (por ejemplo, la paleta codificada con el bloque previamente codificado). Sin embargo, tales paletas pueden estar espacialmente alejadas de la CU actual, lo cual 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 dar como resultado 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.

[0061] En un ejemplo de codificación de paleta, el codificador de vídeo 20 puede generar un elemento sintáctico, tal como un indicador "PLT_Mode_flag", que indica si un modo de codificación basada en paleta se utiliza 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. A continuación, el descodificador de vídeo 30 puede 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 un modo de paletas. En este caso, el descodificador de vídeo 30 puede aplicar el modo de codificación basada en paleta para descodificar la CU. En algunos ejemplos, un elemento sintáctico puede indicar uno de una pluralidad de diferentes modos de paleta para la CU.

[0062] Un valor de este PLT_Mode_flag igual a 0 puede especificar que la CU actual se codifica utilizando un modo 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. El uso de PLT_Mode_flag se describe a modo de ejemplo. Sin embargo, en otros ejemplos, otros elementos sintácticos, tales como los códigos de múltiples bits, se pueden usar para indicar si el modo de codificación basada en paleta se ha de usar para una CU (o PU en otros ejemplos) o para indicar cuáles, entre una pluralidad de modos, deben ser utilizados.

[0063] El PLT_Mode_flag u otro elemento sintáctico 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 sintáctico 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 sintáctico enviado a nivel de fragmento, PPS, SPS o VPS indica que el modo de codificación de paleta está habilitado, el PLT_Mode_flag puede señalarse adicionalmente para indicar si se ha de usar el modo de codificación basada en paleta para cada CU. De nuevo, como se ha mencionado anteriormente, la aplicación de estas técnicas para indicar la codificación basada en paleta de una CU podría usarse, adicional o de forma alternativa, para indicar la codificación basada en paleta de una PU.

[0064] 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.

[0065] 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 sintácticos 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.

5 **[0066]** 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".

20 **[0067]** 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.

[0068] 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 sintácticos 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.

35 **[0069]** 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.

40 **[0070]** 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.

45 **[0071]** 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 sintáctico 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 contiguos causales de la CU actual.

60 **[0072]** 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.

5

[0073] 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 pueden 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".

10

[0074] A continuación, el codificador de vídeo 20 puede generar y señalar una asignación que indica qué respectiva entrada de paleta está asociada con cada píxel en una CU. La *i*-ésima entrada en la asignación 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 obtener 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.

15

20

[0075] 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 sintáctico "run" 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.

25

[0076] 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).

30

35

[0077] En algunos ejemplos, es posible que no sea necesario transmitir la asignación. 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 $1/n, 2/n, \dots, (n-1)/n$ (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.).

40

45

50

[0078] 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.

55

[0079] 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".

60

[0080] 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

65

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.

[0081] 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 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.

[0082] 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.

[0083] Anteriormente, se describió cómo transmitir la asignación identificando el valor de píxel. Junto con el procedimiento descrito anteriormente, la transmisión de la asignación 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). A continuación, 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.

[0084] En el caso de que no se señale 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.

[0085] También es posible que la asignación 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 obtener 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.

[0086] Si se usan ambos procedimientos de transmisión de asignación, entonces un indicador 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".

[0087] La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas de codificación de paleta de esta divulgación. La FIG. 2 se proporciona con propósitos explicativos y no se debería considerar limitativa de las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con propósitos explicativos, esta divulgación describe un codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros estándares o procedimientos de codificación.

[0088] El codificador de vídeo 20 representa un ejemplo de dispositivo que puede estar configurado para realizar técnicas para codificación de vídeo basada en paleta de acuerdo con diversos ejemplos descritos en esta divulgación. Por ejemplo, el codificador de vídeo 20 puede estar configurado para codificar selectivamente varios bloques de datos de vídeo, tales como las CU o PU en la codificación HEVC, usando codificación basada en paleta o codificación no basada en paleta.

[0089] En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de procesamiento de predicción 100, una unidad de generación residual 102, una unidad de procesamiento de transformada 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de procesamiento de transformada inversa 110, una unidad de reconstrucción 112, una unidad de filtro 114, una memoria intermedia de imágenes descodificadas (DPB) 116, una memoria de vídeo 119 y una unidad de codificación por entropía 118. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de predicción inter 120 y una unidad de procesamiento de predicción intra 126. La unidad de procesamiento de predicción inter 120 incluye una unidad de estimación de movimiento y una unidad de compensación de movimiento (no mostradas). El codificador de vídeo 20 también incluye una unidad de codificación basada en paleta 122, configurada para realizar diversos aspectos de las técnicas de codificación basada en paleta descritas en esta divulgación. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales.

[0090] Como se explicará con más detalle a continuación con referencia a las FIGS. 4-9, la unidad de codificación basada en paleta 122 del codificador de vídeo 20 puede configurarse para generar una paleta para un bloque actual de los datos de vídeo, generar un vector de predicción binaria para la paleta para el bloque actual de datos de vídeo, con el vector de predicción binaria que comprende entradas que indican si las entradas de paleta utilizadas anteriormente se reutilizan para la paleta para el bloque actual de datos de vídeo, codificar el vector de predicción binaria usando una técnica de codificación de longitud de ejecución y codificar el bloque actual de datos de vídeo usando la paleta.

[0091] El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede codificar cada CTU en un fragmento de una imagen de los datos de vídeo. Cada una de las CTU puede estar asociada a unos bloques de árbol de codificación (CTB) de luma de igual tamaño y a unos CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede realizar una división 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 unas 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, y así sucesivamente.

[0092] Como se muestra en la FIG. 2, la memoria de vídeo 119 recibe datos de vídeo que se usan para codificar un bloque de vídeo actual dentro de una trama de vídeo. La memoria de vídeo 119 puede almacenar datos de vídeo a codificar por parte de los componentes del codificador de vídeo 20 (por ejemplo, configurarse para almacenar datos de vídeo). Los datos de vídeo almacenados en la memoria de vídeo 119 se pueden obtenerse, por ejemplo, a partir de la fuente de vídeo 18 de la FIG. 1. La DPB 116 es un ejemplo de DPB que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por parte del codificador de vídeo 20 (por ejemplo, en los modos de codificación intra o codificación inter, también denominados modos de codificación de predicción intra o predicción inter). La memoria de vídeo 119 y la DPB 116 pueden estar formadas por cualquiera de entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetoresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de vídeo 119 y la DPB 116 pueden ser proporcionadas por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de vídeo 119 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera del chip con respecto a esos componentes.

[0093] 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 tanto, cada PU puede estar asociada a un bloque de predicción de luma y a unos correspondientes bloques de predicción de croma. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir unas PU que tienen diversos tamaños. Como se ha indicado anteriormente, el tamaño de una CU se puede referir al tamaño del bloque de codificación de luma de la CU, y el tamaño de una PU se puede referir al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular sea de $2N \times 2N$, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir tamaños de PU de $2N \times 2N$ o $N \times N$ para predicción intra, y tamaños de PU simétricas de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares para predicción inter. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden admitir una división asimétrica para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para predicción inter.

[0094] La unidad de procesamiento de predicción inter 120 puede generar datos predictivos para una PU realizando una predicción inter en cada PU de una CU. Los datos predictivos para la PU pueden incluir bloques predictivos de la PU e información de movimiento para la PU. La unidad de predicción inter 121 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, en un fragmento

P o en un fragmento B. En un fragmento I, todas las PU se intrapredicen. Por lo tanto, si la PU está en un fragmento I, la unidad de predicción inter 121 no realiza predicción inter en la PU. Por tanto, para bloques codificados en el modo I, el bloque predicho se forma usando predicción espacial a partir de bloques contiguos previamente codificados dentro de la misma trama.

5

[0095] Si una PU está en un fragmento P, la unidad de estimación de movimiento de la unidad de procesamiento de predicción inter 120 puede buscar en las imágenes de referencia de una lista de imágenes de referencia (por ejemplo, "RefPicList0") una región de referencia para la PU. La región de referencia para la PU puede ser una región, dentro de una imagen de referencia, que contiene bloques de muestras que se corresponden más estrechamente con los bloques de muestras de la PU. La unidad de estimación de movimiento puede generar un índice de referencia que indique una posición en la RefPicList0 de la imagen de referencia que contiene la región de referencia para la PU. Además, la unidad de estimación de movimiento puede generar un MV que indica un desplazamiento espacial entre un bloque de codificación de la PU y una localización de referencia asociada a la región de referencia. Por ejemplo, el MV puede ser un vector bidimensional que proporciona un desplazamiento desde las coordenadas de la imagen descodificada actual hasta las coordenadas de una imagen de referencia. La unidad de estimación de movimiento puede facilitar el índice de referencia y el MV como la información de movimiento de la PU. La unidad de compensación de movimiento de la unidad de procesamiento de predicción inter 120 puede generar los bloques predictivos de la PU basándose en muestras reales o interpoladas en la ubicación de referencia, indicada por el vector de movimiento de la PU.

10

15

20

[0096] 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 uni-predicción para la PU, la unidad de estimación de movimiento puede buscar en las imágenes de referencia de la RefPicList0 o de una segunda lista de imágenes de referencia ("RefPicList1") una región de referencia para la PU. La unidad de estimación de movimiento puede enviar, como la información de movimiento de la PU, un índice de referencia que indica una posición en la RefPicList0 o la RefPicList1 de la imagen de referencia que contiene la región de referencia, un vector de movimiento que indica un desplazamiento espacial entre un bloque de predicción de la PU y una ubicación de referencia asociada a la región de referencia, y uno o más indicadores de dirección de predicción que indican si la imagen de referencia está en la RefPicList0 o la RefPicList1. La unidad de compensación de movimiento de la unidad de procesamiento de predicción inter 120 puede generar los bloques predictivos de la PU basándose, al menos en parte, en muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.

25

30

[0097] Para realizar la predicción inter bidireccional para una PU, la unidad de estimación de movimiento puede buscar las imágenes de referencia en la RefPicList0 para una región de referencia para la PU y también puede buscar las imágenes de referencia en la 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 la RefPicList0 y la RefPicList1 de las imágenes de referencia que contienen las regiones de referencia. Además, la unidad de estimación de movimiento puede generar unos MV que indican desplazamientos espaciales entre la localización de referencia asociada a las regiones de referencia y un bloque de muestras de la PU. La información de movimiento de la PU puede incluir los índices de referencia y los MV de la PU. La unidad de compensación de movimiento puede generar los bloques predictivos de la PU basándose, al menos en parte, en muestras reales o interpoladas en las regiones de referencia indicadas por los vectores de movimiento de la PU.

35

40

[0098] La unidad de procesamiento de predicción intra 126 puede generar datos predictivos para una PU realizando una predicción intra en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos para la PU y varios elementos sintácticos. La unidad de procesamiento de predicción intra 126 puede realizar una predicción intra en las PU en fragmentos I, fragmentos P y fragmentos B.

45

[0099] Para realizar una predicción intra en una PU, la unidad de procesamiento de predicción intra 126 puede usar múltiples modos de predicción intra para generar múltiples conjuntos de datos predictivos para la PU. La unidad de procesamiento de predicción intra 126 puede usar muestras de bloques de muestra de las PU contiguas para generar un bloque predictivo para una PU. Las PU contiguas pueden estar encima, encima y a la derecha, encima y a la izquierda, o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha y de arriba abajo, para las PU, CU y CTU. La unidad de procesamiento de predicción intra 126 puede usar diversos números de modos de predicción intra, por ejemplo, 33 modos de predicción intra direccional. En algunos ejemplos, el número de modos de predicción intra puede depender del tamaño de la región asociada a la PU.

50

55

[0100] La unidad de procesamiento de predicción 100 puede seleccionar los datos predictivos para las PU de una CU de entre los datos predictivos generados por la unidad de procesamiento de predicción inter 120 para las PU, o los datos predictivos generados por la unidad de procesamiento de predicción intra 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 unas métricas de velocidad/distorsión de los conjuntos de datos predictivos. Los bloques predictivos de los datos predictivos seleccionados pueden denominarse en el presente documento bloques predictivos seleccionados.

60

65

[0101] La unidad de generación residual 102 puede generar, basándose en el bloque 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, un bloque residual (por ejemplo, bloque residual de luma, Cb y Cr) de la CU. Por ejemplo, la unidad de generación residual 102 puede generar los bloques residuales de la CU de manera que cada muestra en los bloques residuales tenga un valor igual a una diferencia entre una muestra en un bloque de codificación de la CU y una muestra correspondiente en un correspondiente bloque predictivo seleccionado de una PU de la CU.

[0102] La unidad de procesamiento de transformada 104 puede realizar una división de árbol cuaternario para dividir los bloques residuales asociados a una CU en bloques de transformada asociados a unas TU de la CU. De este modo, una TU puede corresponder a un bloque de transformada de luma y dos bloques de transformada de croma. Los tamaños y las posiciones de los bloques de transformada de luma y croma de unas TU de una CU pueden o no estar basados en los tamaños y las posiciones de los bloques de predicción de las PU de la CU. Una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT) puede incluir nodos correspondientes a cada una de las regiones. Las TU de una CU pueden corresponder a nodos hoja del RQT.

[0103] La unidad de procesamiento de transformada 104 puede generar bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas a un bloque de transformada asociado a una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar al bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 104 no aplica transformadas a un bloque de transformada. En dichos ejemplos, el bloque de transformada se puede tratar como un bloque de coeficientes de transformada.

[0104] La unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de n bits puede redondearse hacia abajo hasta un coeficiente de transformada de m bits durante la cuantificación, donde n es mayor que m . La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado a una TU de una CU basándose en un valor de parámetro de cuantificación (QP) asociado a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados a una CU, ajustando el valor de QP asociado a la CU. La cuantificación puede introducir pérdida de información, por tanto unos coeficientes de transformada cuantificados pueden tener una precisión menor que los originales.

[0105] La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformada inversa 110 pueden aplicar una cuantificación inversa y transformadas 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 generados por la unidad de procesamiento de predicción 100 para producir un bloque de transformada reconstruido asociado a una TU. Reconstruyendo bloques de transformada para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

[0106] La unidad de filtro 114 puede realizar una o más operaciones antibloque para reducir los artefactos de bloque en los bloques de codificación asociados a 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 la una o más operaciones antibloque en los bloques de codificación reconstruidos. La unidad de procesamiento de predicción inter 120 puede usar una imagen de referencia que contiene los bloques de codificación reconstruidos para realizar una predicción inter en las PU de otras imágenes. Además, la unidad de procesamiento de predicción intra 126 puede usar bloques de codificación reconstruidos en la memoria intermedia de imágenes descodificadas 116 para realizar una predicción intra en otras PU de la misma imagen que la CU.

[0107] 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 sintácticos 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 sometidos a codificación por entropía. Por ejemplo, la unidad de codificación por entropía 118 puede realizar una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de división de intervalos de probabilidad (PIPE), una operación de codificación Golomb exponencial u otro tipo de operación de codificación por entropía en los datos. El codificador de vídeo 20 puede facilitar una corriente de bits que incluye datos sometidos a codificación por entropía, generados por la unidad de codificación por entropía 118. Por ejemplo, la corriente de bits puede incluir datos que representan un RQT para una CU.

[0108] La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que está configurado para implementar las técnicas de esta divulgación. La FIG. 3 se proporciona con propósitos explicativos y no se limita a las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con propósitos explicativos, esta divulgación describe un descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros estándares o procedimientos de codificación.

[0109] En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye 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 transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 160, una memoria de vídeo 163 y una memoria intermedia de imágenes descodificadas (DPB) 162. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 164 y una unidad de procesamiento de predicción intra 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 funcionales.

[0110] Como se explicará con más detalle a continuación con referencia a las FIGS. 4-9, la unidad de descodificación basada en paleta 165 del descodificador de vídeo 30 puede configurarse para recibir un vector de predicción binaria codificada (por ejemplo, un vector de predicción binaria codificada con codificación de longitud de ejecución) para un bloque actual de los datos de vídeo, descodificar el vector de predicción binaria codificada que utiliza una técnica de descodificación de longitud de ejecución, generar una paleta para el bloque actual de datos de vídeo basándose en el vector de predicción binaria, con el vector de predicción binaria que comprende entradas que indican si las entradas de paleta utilizadas previamente se reutilizan para la paleta para el bloque actual de datos de vídeo y descodificar el bloque actual de datos de vídeo usando la paleta.

[0111] Una memoria intermedia de imágenes codificadas (CPB), por ejemplo, memoria de vídeo, 163 puede recibir y almacenar datos de vídeo codificados (por ejemplo, unidades NAL) de un flujo de bits. La unidad de descodificación por entropía 150 puede recibir datos de vídeo codificados (por ejemplo, unidades NAL) desde la CPB y analizar sintácticamente las unidades NAL para descodificar elementos sintácticos. La unidad de descodificación por entropía 150 puede realizar la descodificación por entropía de unos elementos sintácticos sometidos a codificación por entropía en las unidades NAL. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de procesamiento de transformada inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 160 pueden generar datos de vídeo descodificados basándose en los elementos sintácticos extraídos de la corriente de bits.

[0112] Como se muestra en la FIG. 3, la memoria de vídeo 163 recibe datos de vídeo que se usan para descodificar un bloque de vídeo actual dentro de una trama de vídeo. La memoria de vídeo 163 puede almacenar datos de vídeo para ser descodificados por los componentes del descodificador de vídeo 30 (por ejemplo, configurarse para almacenar datos de vídeo). Los datos de vídeo almacenados en la memoria de vídeo 163 se pueden obtener, por ejemplo, a partir de un flujo de vídeo codificado producido por el codificador de vídeo 20. La DPB 162 es un ejemplo de DPB, que almacena datos de vídeo de referencia para su uso en la descodificación de datos de vídeo mediante el descodificador de vídeo 30 (por ejemplo, en los modos de codificación intra o inter, también denominados modos de codificación de predicción intra o inter). La memoria de vídeo 163 y la DPB 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 magnetoresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de vídeo 163 y la DPB 162 pueden ser proporcionadas por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de vídeo 163 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera del chip con respecto a esos componentes.

[0113] Las unidades NAL de la corriente de bits pueden incluir unidades NAL de fragmentos codificados. Como parte de la descodificación de la corriente de bits, la unidad de descodificación por entropía 150 puede extraer y realizar la descodificación por entropía unos elementos sintácticos de las unidades NAL de fragmentos codificados. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos sintácticos pertenecientes a un fragmento. Los elementos sintácticos de la cabecera de fragmento pueden incluir un elemento sintáctico que identifica un PPS asociado a una imagen que contiene el fragmento.

[0114] Además de descodificar los elementos sintácticos de la corriente de bits, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Realizando la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir bloques residuales de la CU.

- 5 [0115] Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede realizar la cuantificación inversa, es decir, la descuantificación, de los bloques de coeficientes asociados a la TU. La unidad de cuantificación inversa 154 puede usar 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 que la unidad de cuantificación inversa 154 lo aplique. Es decir, la relación de compresión, es decir, la relación entre el número de bits usados para representar la secuencia original y la comprimida, puede controlarse ajustando el valor del QP usado al cuantificar los coeficientes de transformación. La relación de compresión también puede depender del procedimiento de codificación por entropía empleado.
- 10 [0116] Después de que la unidad de cuantificación inversa 154 haya realizado la cuantificación inversa de un bloque de coeficientes, la unidad de procesamiento de transformada inversa 156 puede aplicar una o más transformadas inversas al bloque de coeficientes a fin de generar un bloque residual asociado a la TU. Por ejemplo, la unidad de procesamiento de transformada inversa 156 puede aplicar una DCT inversa, una transformada entera inversa, una transformada inversa de Karhunen-Loeve (KLT), una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes.
- 15 [0117] Si se codifica una PU usando predicción intra, la unidad de procesamiento de predicción intra 166 puede realizar una predicción intra para generar bloques predictivos para la PU. La unidad de procesamiento de predicción intra 166 puede usar un modo de predicción intra para generar los bloques predictivos de luma, Cb y Cr para la PU basándose en los bloques de predicción de unas PU contiguas en el espacio. La unidad de procesamiento de predicción intra 166 puede determinar el modo de predicción intra para la PU basándose en uno o más elementos sintácticos descodificados de la corriente de bits.
- 20 [0118] 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 sintácticos extraídos del flujo de bits. Además, si se codifica una PU usando predicción inter, la unidad de descodificación por entropía 150 puede extraer información de movimiento para la PU. La unidad de compensación de movimiento 164 puede determinar, basándose en la información de movimiento de la PU, una o más regiones de referencia para la PU. La unidad de compensación de movimiento 164 puede generar, basándose en bloques de muestras en uno o más bloques de referencia para la PU, bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para la PU.
- 25 [0119] La unidad de reconstrucción 158 puede usar los bloques de transformada (por ejemplo, los bloques de transformada de luma, Cb y Cr) asociados con las TU de una CU y los bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) de las PU de la CU, es decir, cualquiera de los datos de predicción intra o de los datos de predicción inter, 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 transformada (por ejemplo, los bloques de transformada de luma, Cb y Cr) a las muestras 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.
- 30 [0120] 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 una posterior compensación de movimiento, predicción intra y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en los bloques (por ejemplo, los bloques de luma, Cb y Cr) en la memoria intermedia de imágenes descodificadas 162, operaciones de predicción intra o de predicción inter en las PU de otras CU.
- 35 [0121] La FIG. 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 un vector de predicción binaria.
- 40 [0122] 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
- 45
- 50
- 55
- 60
- 65

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.

[0123] Una vez que una paleta es generada por la unidad de generación de paleta 203, la unidad de asignación 204 puede generar una asignación 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 asignación 204 puede producir una asignación 214 que incluye elementos sintácticos que indican cómo cada píxel usa (o no usa) las entradas de la paleta. Si el valor de un píxel en el bloque actual de datos de vídeo no se encuentra en la paleta y, por lo tanto, no puede representarse mediante un índice en la paleta, la unidad de asignación 204 puede transmitir explícitamente un valor de píxel para ese píxel en particular. En algunos ejemplos, la unidad de asignación 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 asignación 204 puede cuantificar el píxel y transmitir los valores cuantificados.

[0124] Además de la señalización de los elementos sintácticos 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 señalan para indicar los valores de una paleta para un bloque particular de datos de vídeo.

[0125] 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 predictor. Una paleta de predictor puede incluir entradas de paleta de bloques previamente codificados que usan el modo de paleta u otras muestras reconstruidas. Como se muestra en la FIG. 4, la unidad de codificación basada en paleta 122 puede incluir una memoria intermedia de paleta de predictor 210. La memoria intermedia de paleta de predictor 210 puede configurarse para almacenar un número de entradas de la paleta utilizadas previamente de bloques previamente codificados. Como ejemplo, la memoria intermedia de paleta de predictor 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 predictor 210 puede ser de cualquier tamaño. En un ejemplo, la memoria intermedia de paleta de predictor 210 incluye hasta 64 entradas de paleta utilizadas anteriormente.

[0126] En algunos ejemplos, la unidad de codificación basada en paleta 122 puede configurarse para eliminar las entradas en la memoria intermedia de paleta de predictor 210, de modo que todas las entradas de la paleta en la memoria intermedia de paleta de predictor 210 son únicas. Es decir, para cada nueva entrada de paleta que se agregue a la memoria intermedia de paleta de predictor 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 predictor 210. Si no hay entradas idénticas, la nueva entrada de la paleta se agrega a la memoria intermedia de paleta de predictor 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 predictor 210 y las entradas duplicadas se eliminan de la memoria intermedia de paleta de predictor 210.

[0127] La unidad de codificación basada en paleta 122 puede incluir una unidad de generación de vectores de predicción binaria 206 que está configurada para generar y señalar un indicador binario 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 predictor 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 binarias indica que la entrada correspondiente en la memoria intermedia de paleta de predictor 210 se reutiliza para la paleta para el bloque actual, mientras que un indicador con un valor de 0 en el vector de predicción binarias indica que la entrada correspondiente en la memoria intermedia de paleta de predictor 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 predictor 210. El número de nuevas entradas también puede ser señalado.

[0128] En la Solicitud Provisional de los Estados Unidos n.º 61/970 257, presentada el 25 de marzo de 2014; la Solicitud Provisional de los Estados Unidos n.º 61/981 105, presentada el 17 de abril de 2014; y la Solicitud Provisional de los Estados Unidos n.º 62/002 668, presentada el 23 de mayo de 2014, se propuso un procedimiento de señalización basado en árbol binario y procedimientos de señalización basados en posición final para la codificación del vector predictor binario de paleta. En la Solicitud Provisional de Estados Unidos n.º 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 descodificar el vector de predicción binaria.

[0129] 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 binaria generado por la unidad de generación del vector de predicción binaria 206 se denota por:

5

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

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 binaria). Si $N=0$, $\mathbf{b} =$ (es decir, \mathbf{b} es el vector vacío), que no necesita ser señalado. Por lo tanto, en la siguiente descripción, se puede suponer que $N > 0$.

10

[0130] La FIG. 5 muestra un ejemplo de una memoria intermedia de paleta de predictor 210 y una paleta 220 actual. Como puede verse en la FIG. 5, la paleta 220 actual reutiliza los valores de pixel de la memoria intermedia de la paleta de predictor 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 binaria 206 de la FIG. 4 sería $\mathbf{b} = [110010001000]$. Como se puede ver en este ejemplo, el vector de predicción binaria \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 predictor 210. Es decir, la 1.ª, 2.ª, 5.ª y 9.ª entradas en la memoria intermedia de paleta de predictor 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).

15

20

[0131] 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 FIG. 4, la unidad de compresión del vector de predicción binaria 209 puede configurarse para generar y señalar el vector de predicción binaria 215 codificado. Sin embargo, debe entenderse que las técnicas de compresión del vector de predicción binaria de esta divulgación pueden implementarse en otras estructuras del codificador de vídeo 20, incluida la unidad de codificación por entropía 118.

25

30

[0132] En un ejemplo de la divulgación, la unidad de compresión del vector de predicción binaria 209 puede configurarse para codificar el vector de predicción binaria utilizando técnicas de codificación basadas en longitud de ejecución. Por ejemplo, la unidad de compresión del vector de predicción binaria 209 puede configurarse para codificar el vector de predicción binaria mediante la señalización del número de "0" consecutivos entre "1" en el vector de predicción binaria utilizando un código exponencial-Golomb. Como ejemplo, supongamos nuevamente que $\mathbf{b} = [110010001000]$. En este ejemplo, como se muestra en la FIG. 6, el vector de predicción binaria (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 binaria 209 puede utilizar técnicas de codificación por longitud de ejecución basada en cero para representar el vector de predicción binaria \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".

35

40

45

[0133] 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 binaria 209 utiliza un código de Exponencial-Golomb de orden 0 como la técnica de codificación de longitud de ejecución.

50

[0134] Para la binarización truncada, el símbolo max puede ser el valor máximo posible de la ejecución dependiendo 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, el símbolo max puede ser la longitud del vector binario o la longitud del vector binario menos la posición del "1" desde la 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 el símbolo max se encuentra entre corchetes.

55

60

[0135] 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.

65

- 5 [0136] 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 señalarse 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 puede configurarse para detener la señalización de longitud de ejecución cuando el número de elementos "1" señalados (es decir, el número de entradas de la paleta desde la memoria intermedia de paleta de predictor 210 indicado como reutilizado para la paleta actual 220) alcanza un número máximo posible. En algunos ejemplos, el número máximo posible es el tamaño máximo posible de la paleta.
- 10 [0137] 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 binaria \mathbf{b} . En uno o más ejemplos de esta divulgación, la unidad de compresión del vector de predicción binaria 209 puede configurarse para codificar el vector de predicción binaria \mathbf{b} usando una longitud de ejecución reservada L para codificar la posición final del vector de predicción binaria. 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 binaria 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 binaria 209 está configurada para señalar la longitud de ejecución tal como está. La unidad de compresión del vector de predicción binaria 209 puede señalar la longitud de ejecución de la posición final con la longitud de ejecución reservada L .
- 15 [0138] 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 binaria \mathbf{b} son todas 0. Por lo tanto, si el valor descodificado es igual a L , no es necesaria más señalización de ejecución.
- 20 [0139] Utilizando el mismo ejemplo anterior (es decir, $\mathbf{b} = [110010001000]$) y suponiendo que $L=1$, la unidad de compresión del vector de predicción binaria 209 está configurada para señalar la secuencia de longitud de ejecución "0-0-2-3-4" de la FIG. 6 como "0-0-3-4-1". A continuación, 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-end". 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 binaria.
- 25 [0140] En otro ejemplo de la divulgación, la unidad de compresión del vector de predicción binaria 209 puede configurarse para aplicar solo codificación de posición final cuando el número total de "1" en el vector de predicción binaria (es decir, el número de indicaciones de entradas de paleta reutilizadas desde la memoria intermedia de paleta de predictor 210) es menor que el tamaño máximo de paleta posible. Si el número total de "1" en el vector de predicción binaria es igual a un tamaño máximo de paleta posible, la unidad de compresión del vector de predicción binaria 209 puede configurarse para obtener la señalización de la última longitud de ejecución.
- 30 [0141] En el ejemplo anterior, si el descodificador de vídeo 30 determina que el tamaño máximo de paleta posible es 4, entonces una secuencia de duración de ejecución de '0-0-2-3' (o secuencia de duración de ejecución 0-0-3-4 de acuerdo con las reglas de longitud de ejecución reservadas descritas anteriormente) puede ser suficiente para recuperar el vector \mathbf{b} de predicción binaria. En algunos ejemplos, el tamaño máximo de paleta posible puede estar predeterminado. En consecuencia, el tamaño máximo de paleta posible se puede determinar de antemano por el codificador de vídeo 20 y el descodificador de vídeo 30. En otros ejemplos, el tamaño máximo de paleta posible puede señalarse mediante el codificador de vídeo 20 en el flujo de bits de vídeo codificado. La señalización del tamaño máximo de paleta posible puede realizarse usando cualquier técnica de comunicación de datos convencional adecuada o cualquier técnica de comunicación de datos adecuada descrita en el presente documento.
- 35 [0142] En otro ejemplo de la divulgación, la unidad de compresión del vector de predicción binaria 209 puede configurarse para aplicar las técnicas de codificación de posición final descritas anteriormente solo cuando la última longitud de ejecución en la secuencia de longitud de ejecución no es 0. Si la última longitud de ejecución en la secuencia de longitud de ejecución es 0, se puede obtener la codificación de posición final. Esta declaración es equivalente a cuando el último indicador en el vector de predicción binaria \mathbf{b} es igual a 1. En esta situación, se puede obtener la codificación de la posición final. Por ejemplo, si $\mathbf{b} = [100000000001]$, entonces la secuencia de longitud de ejecución es '0-10-0'. La secuencia de longitud de ejecución de '0-10' (o secuencia de valor señalada
- 40
- 45
- 50
- 55
- 60
- 65

0-11 de acuerdo con las reglas de longitud de ejecución reservadas descritas anteriormente) puede ser suficiente para recuperar el vector de predicción binaria **b**. Además, la posición final puede señalarse sin indicar ninguna ejecución para indicar que el vector de predicción binaria **b** es un vector cero, lo que significa que no se predicen entradas de paleta.

5

[0143] En algunos ejemplos, en lugar de usar un valor fijo L para indicar que los indicadores binarios restantes son 0, se puede usar un mínimo de L y el tamaño del vector predictor binario para indicar la posición final. Esto se debe a que el valor de la ejecución siempre es menor que el tamaño del vector predictor binario.

10

[0144] La FIG. 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 FIG. 4. La unidad de descodificación basada en paleta 165 puede configurarse para recibir una asignación 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 asignación 312 puede indicar además qué entradas de la paleta se utilizarán para un píxel dado. La unidad de asignación 302 puede descodificar el bloque actual de datos de vídeo usando la asignación 312 y una paleta generada por la unidad de generación de paletas 304 para producir datos de vídeo descodificados 314.

15

20

[0145] De acuerdo con las técnicas de esta divulgación, la unidad de descodificación basada en paleta 165 también puede recibir un vector de predicción binaria codificada 316. Como se analizó anteriormente, el vector de predicción binaria 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 binaria. La unidad de descompresión del vector de predicción binaria 306 puede configurarse para descodificar el vector de predicción binaria codificada usando cualquier combinación de las técnicas de codificación de longitud de ejecución descritas anteriormente con referencia a las FIGS. 4-6. Una vez que se recupera un vector de predicción binaria mediante la unidad de descompresión del vector de predicción binaria 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 binaria y las entradas de la paleta utilizadas previamente almacenadas en la memoria intermedia de paletas de predictor 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 predictor 310 de la misma manera que la unidad de codificación basada en paleta 122 almacenó las entradas de la paleta usadas previamente en la memoria intermedia de paleta de predictor 210.

25

30

35

[0146] La FIG. 8 es un diagrama de flujo que ilustra un procedimiento de codificación de ejemplo de la divulgación. Las técnicas de la FIG. 8 puede implementarse mediante uno o más componentes estructurales del codificador de vídeo 20, incluida la unidad de codificación basada en paleta 122. En un ejemplo de la divulgación, la unidad de codificación de paleta 122 puede configurarse para generar una paleta para el bloque actual de datos de vídeo (800), y generar un vector de predicción binaria para la paleta para el bloque actual de datos de vídeo (802). El vector de predicción binaria comprende entradas que indican si las entradas de paleta utilizadas previamente se reutilizan para la paleta para el bloque actual de datos de vídeo. La unidad de codificación basada en paleta 122 puede configurarse adicionalmente para codificar el vector de predicción binaria usando una técnica de codificación de longitud de ejecución (804), y codificar el bloque actual de datos de vídeo usando la paleta (806). En un ejemplo, la técnica de codificación de longitud de ejecución comprende codificar una longitud de ejecución de ceros.

40

45

[0147] En un ejemplo de la divulgación, la unidad de codificación basada en paleta 122 puede configurarse para codificar el vector de predicción binaria utilizando una técnica de codificación exponencial-Golomb. En un ejemplo, la técnica de codificación exponencial-Golomb es una técnica de codificación exponencial-Golomb de orden 0.

50

[0148] En otro ejemplo de la divulgación, la unidad de codificación basada en paleta 122 puede configurarse para codificar el vector de predicción binaria usando la técnica de codificación de longitud de ejecución y un valor de longitud de ejecución reservado L , con el valor de longitud de ejecución reservado L que indica una posición final del vector de predicción binaria. En un ejemplo, el valor de longitud de ejecución reservado L es 1.

55

[0149] En otro ejemplo de la divulgación, la unidad de codificación basada en paleta 122 puede configurarse para codificar el vector de predicción binaria usando la técnica de codificación de longitud de ejecución, un valor de longitud de ejecución reservado L y un tamaño máximo de paleta. El valor de longitud de ejecución reservado L indica una posición final del vector de predicción binaria. En este ejemplo, el valor de longitud de ejecución reservado L no se usa si un número total de entradas en el vector de predicción binaria que indica que las entradas de paleta usadas previamente se reutilizan para la paleta para el bloque actual de datos de vídeo es igual al tamaño máximo de paleta.

60

65

[0150] En otro ejemplo de la divulgación, la unidad de codificación basada en paleta 122 puede configurarse para codificar el vector de predicción binaria usando la técnica de codificación de longitud de ejecución, y un valor de longitud de ejecución reservado L , con el valor de longitud de ejecución reservado L que indica una posición final del vector de predicción binaria. En este ejemplo, el valor de longitud de ejecución reservado L no se usa si

una última longitud de ejecución en el vector de predicción binaria codificada no indica una longitud de ejecución de cero.

5 **[0151]** En otro ejemplo de la divulgación, la unidad de codificación basada en paleta 122 puede configurarse para almacenar entradas de paleta usadas previamente para uno o más bloques de datos de vídeo previamente codificados en la memoria intermedia. La unidad de codificación basada en paleta 122 puede configurarse adicionalmente para eliminar entradas duplicadas de las entradas de paleta usadas previamente almacenadas en la memoria intermedia. En otro ejemplo de la divulgación, las entradas de paleta usadas previamente para uno o más bloques de datos de vídeo previamente codificados comprenden entradas de paleta usadas previamente para una línea de píxeles sobre el bloque actual de datos de vídeo y entradas de paleta usadas previamente para una línea de píxeles a la izquierda del bloque actual de datos de vídeo.

15 **[0152]** La FIG. 9 es un diagrama de flujo que ilustra un procedimiento de descodificación de ejemplo de la divulgación. Las técnicas de la FIG. 9 pueden implementarse mediante uno o más componentes estructurales del descodificador de vídeo 30, incluyendo la unidad de descodificación basada en paleta 165.

20 **[0153]** En un ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse para recibir un vector de predicción binaria codificada para un bloque actual de datos de vídeo (900), y descodificar el vector de predicción binaria codificada utilizando una técnica de descodificación de longitud de ejecución (902). En un ejemplo, la técnica de descodificación de longitud de ejecución comprende codificar una longitud de ejecución de ceros. La unidad de descodificación basada en paleta 165 puede configurarse adicionalmente para generar una paleta para el bloque actual de datos de vídeo basándose en el vector de predicción binaria (904). El vector de predicción binaria que comprende entradas que indican si las entradas de paleta utilizadas anteriormente se reutilizan para la paleta para el bloque actual de datos de vídeo. La unidad de descodificación basada en paleta 25 165 puede configurarse adicionalmente para descodificar el bloque actual de datos de vídeo usando la paleta (906).

30 **[0154]** En otro ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse adicionalmente para descodificar el vector de predicción binaria codificada usando una técnica de descodificación exponencial-Golomb. En un ejemplo, la técnica de descodificación exponencial-Golomb es una técnica de descodificación exponencial-Golomb de orden 0.

35 **[0155]** En otro ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse adicionalmente para descodificar el vector de predicción binaria codificada usando la técnica de descodificación de longitud de ejecución y un valor de longitud de ejecución reservado L, con el valor de longitud de ejecución reservado L que indica una posición final del vector de predicción binaria. En un ejemplo, el valor de longitud de ejecución reservado L es 1.

40 **[0156]** En otro ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse adicionalmente para descodificar el vector de predicción binaria codificada usando la técnica de descodificación de longitud de ejecución, un valor de longitud de ejecución reservado L y un tamaño máximo de paleta. El valor de longitud de ejecución reservado L indica una posición final del vector de predicción binaria. El valor de longitud de ejecución reservado L no se usa si un número total de entradas en el vector de predicción binaria que indica que las entradas de paleta usadas previamente se reutilizan para la paleta para el bloque actual de datos de vídeo es 45 igual al tamaño máximo de paleta.

50 **[0157]** En otro ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse adicionalmente para descodificar el vector de predicción binaria codificada usando la técnica de descodificación de longitud de ejecución y un valor de longitud de ejecución reservado L, con el valor de longitud de ejecución reservado L que indica una posición final del vector de predicción binaria. El valor de longitud de ejecución reservado L no se utiliza si una última longitud de ejecución en el vector de predicción binaria codificada no indica una longitud de ejecución de cero.

55 **[0158]** En otro ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse adicionalmente para copiar, desde una memoria intermedia, entradas de paleta usadas previamente en la paleta que se indican como reutilizadas para la paleta por parte del vector de predicción binaria, y recibir, en el caso de que el número de entradas de paleta utilizadas previamente copiadas para la paleta sea inferior al tamaño máximo de la paleta, entradas de paleta adicionales.

60 **[0159]** En otro ejemplo de la divulgación, la unidad de descodificación basada en paleta 165 puede configurarse adicionalmente para almacenar entradas de paleta usadas previamente para uno o más bloques de datos de vídeo previamente descodificados en la memoria intermedia. La unidad de descodificación basada en paleta 165 puede configurarse adicionalmente para eliminar entradas duplicadas de las entradas de paleta usadas previamente almacenadas en la memoria intermedia. En un ejemplo, las entradas de paleta usadas previamente para el uno o 65 más bloques de datos de vídeo previamente descodificados comprenden entradas de paleta usadas previamente

para una línea de píxeles sobre el bloque actual de datos de vídeo y entradas de paleta usadas previamente para una línea de píxeles a la izquierda del bloque actual de datos de vídeo.

[0160] Las siguientes secciones describen ejemplos de técnicas adicionales de la divulgación. En un primer ejemplo, el codificador de vídeo 20 puede configurarse para señalar primero el número total de unidades ("número de uno"), es decir, el número de entradas de paleta reutilizadas de la memoria intermedia de paletas predictoras 210, en el vector de predicción binaria \mathbf{b} usando un código Golomb-Rice, un código Exponencial-Golomb, o un código Truncado-Rice, o su combinación. A continuación, una vez que el número de indicadores de predicción binaria no cero descodificada alcanza el "número de uno" señalado, el descodificador de vídeo 30 puede determinar que los indicadores restantes en el vector de predicción binaria \mathbf{b} son 0 y la señalización de estos indicadores fue ignorada por el codificador de vídeo 20. En otras palabras, en este primer ejemplo, se puede obtener el procedimiento de codificación de posición final descrito anteriormente. En este caso, se indican las longitudes de ejecución reales. En tal ejemplo, no es necesario un ajuste ascendente a los valores de longitud de ejecución.

[0161] En un segundo ejemplo, como se describió anteriormente, solo la longitud de ejecución de los valores cero en el vector de predicción binaria es código. Como técnica alternativa, la codificación de longitud de ejecución puede incluir (por ejemplo, permitir) ambas longitudes de ejecución de valores cero en el vector de predicción binaria y/o longitudes de ejecución de uno de los valores en el vector de predicción binaria. La ejecución inicial se puede predefinir para que sea una 'longitud de ejecución de uno' (o "longitud de ejecución de cero"). A continuación, el codificador de vídeo 20 puede alternar la codificación de las series de ceros y series de unos.

[0162] Por ejemplo, utilizando el mismo ejemplo que el anterior (es decir, $\mathbf{b} = [110010001000]$) y suponiendo que la primera ejecución es "longitud de ejecución de uno", $\mathbf{b} = [1100100010000]$ puede expresarse como:

"Run=2"- "Run=2"- "Run=12"- "Run=3"- "Run=1"- "Run=4" Sin embargo, se sabe que "ejecución cero" debe ir seguido de "ejecución uno" y viceversa; por lo tanto, la expresión anterior se puede simplificar como:

"Run=2"- "Run=1"- "Run=0"- "Run=2"- "Run=0"- "Run=3" que puede expresarse como secuencia de longitud de ejecución como "210203". La codificación Golomb-Rice, la codificación exponencial-Golomb o la codificación Truncated-Rice pueden usarse para codificar esta secuencia. La "Ejecución cero" y la "Ejecución uno" pueden tener diferentes parámetros de un código Golomb-Rice, un código exponencial-Golomb o un código Truncated-Rice. Se pueden usar diferentes contextos para "Ejecución cero" y "Ejecución uno" si se usan bins regulares para codificarlos. Las técnicas de codificación de posición final descritas en otra parte de esta divulgación también pueden aplicarse en este ejemplo.

[0163] Además, el codificador de vídeo 20 puede configurarse para detener la señalización de longitud de ejecución cuando el número de elementos "1" señalados alcanza un número máximo posible, que puede ser el tamaño máximo de paleta posible.

[0164] En un tercer ejemplo, b_0 , el primer elemento en el vector de predicción binaria \mathbf{b} , puede señalarse directamente para que el descodificador de vídeo 30 pueda determinar si la primera ejecución es "ejecución cero" o "ejecución uno" basándose en el valor de descodificación b_0 .

[0165] En un cuarto ejemplo, el codificador de vídeo 20 puede configurarse para codificar los primeros indicadores M , es decir, $[b_0, b_1, \dots, b_{M-1}]$ en el vector de predicción binaria usando una primera técnica de codificación y el resto, es decir, $[b_M, b_{M+1}, \dots, b_{N-1}]$ se codifican utilizando una segunda técnica de codificación. La segunda técnica de codificación es el procedimiento basado en la longitud de ejecución especificado anteriormente. En un ejemplo, $M = 4$. La primera técnica de codificación puede ser cualquier técnica de codificación convencional adecuada. En algunos ejemplos, la primera técnica de codificación puede ser un procedimiento basado en codificación de longitud no ejecutada. En otros ejemplos, las técnicas de codificación primera y segunda pueden ser técnicas de codificación basadas en la longitud de la ejecución, pero la primera técnica de codificación y la segunda técnica de codificación pueden ser diferentes entre sí.

[0166] En un ejemplo de señalización basada en la longitud de ejecución para ambos elementos binarios, como se ha mencionado anteriormente, el codificador de vídeo 20 puede configurarse para señalar la longitud de ejecución m para "0" o "1" para el vector de predicción binaria \mathbf{b} . Sin embargo, los dos pueden combinarse cuando la longitud de ejecución se señala para ambos elementos binarios. Esto se puede conseguir mediante los siguientes pasos: (1) al principio, se selecciona un valor bin , por ejemplo, $bin = 0$ (2) señalar la longitud de ejecución para el elemento bin , (3) si bin es igual a 1, señalar el *último* indicador que indica si el bin igual a 1 es el último elemento "1" en el vector binario. Por ejemplo, el *último* igual a 1 indica que es el último '1' y, de lo contrario, igual a 0. Si el *último* es igual a 1, dejar de señalar elementos vectoriales binarios y salir; (4) el valor de bin se intercambia, es decir, $bin = 1 - bin$ e ir al paso (2) anterior.

[0167] Si el primer valor de bin se selecciona como 0, entonces puede ser necesario indicar el caso cuando el vector binario es cero. Esta divulgación puede referirse a un indicador que indica si el vector binario es cero como

zero_flag. Por ejemplo, se puede usar un indicador separado al comienzo de la señalización de longitud de ejecución. El indicador separado puede ser igual a 1, lo cual indica que el vector binario es cero. En este caso, no hay señalización de longitud de ejecución después de zero_flag. Si zero_flag es 0, se indican los valores de longitud de ejecución. Si se elige el primer *bin* para que sea igual a 1, entonces puede que no sea necesario señalar este zero_flag, porque el *bin* igual a 1 tiene el last_flag que indica el final de la señalización.

[0168] En un ejemplo, el algoritmo básico analizado anteriormente puede mejorarse aún más usando una o más de las técnicas en la siguiente lista. Cada elemento de la lista puede denominarse "elemento mejorado" en la descripción que sigue a continuación de esta lista:

1. La selección de *bin* puede fijarse en 0 o 1, o ser adaptativa (es decir, ser diferente para varios bloques). Por ejemplo, *bin* se puede elegir (es decir, seleccionar) basándose en los bloques anteriores al identificar qué elemento es el más frecuente o qué primer elemento es el más frecuente en los bloques previamente codificados. De forma alternativa, el *bin* se puede indicar para cada bloque, o se puede indicar para un grupo de bloques (por ejemplo, para cada elemento o en la cabecera del fragmento). De forma alternativa, el *bin* puede señalarse por imagen en una secuencia de vídeo, por ejemplo, en al menos uno del conjunto de parámetros de imagen (PPS), conjunto de parámetros de secuencia (SPS), conjunto de parámetros de vídeo (VPS) o en cualquier otro lugar.

2. El último indicador no puede señalarse si el elemento del vector binario "1" para el cual se señala este último indicador es el último elemento en el vector binario *b*.

3. Después de procesar el primer elemento en el vector binario, el valor de longitud de ejecución señalado puede ser el valor de longitud de ejecución real menos 1, porque si el elemento no es el último, entonces se sabe que es seguido por al menos otro elemento binario.

4. El valor de longitud de ejecución para el último elemento "1" en el vector binario no puede señalarse si el valor de longitud de ejecución para el último elemento "1" en el vector binario está precedido por el elemento "0", ya que si el valor de longitud de ejecución "0" se detiene antes del último elemento en el vector binario, se sabe que este elemento es "1" y es el último elemento "1" en el vector.

5. De forma alternativa, a la señalización zero_flag, el valor de longitud de ejecución para el último elemento "1" en el vector binario puede incluirse en el primer valor de longitud de ejecución donde se asigna un valor específico para esto. Por ejemplo, si el primer valor de longitud de ejecución es cero, el vector binario es cero y no hay otra señalización después de eso. Para el vector binario distinto de cero, el primer valor de longitud de ejecución señalado debe incrementarse en 1 después de ese valor específico, 0 en este ejemplo, para evitar la ambigüedad.

6. El zero_flag y el último indicador pueden ser codificados por contexto de codificación aritmética binaria (CABAC) adaptable al contexto, donde, por ejemplo, se puede asignar un contexto separado para cada indicador. Los contextos pueden depender del tamaño del bloque, el tipo de fragmento, etc.

7. Los valores de longitud de ejecución se pueden codificar utilizando varios procedimientos de binarización, como código unario, código de Golomb-Rice o Exponencial-Golomb, o versiones truncadas de los códigos teniendo en cuenta que el valor de longitud de ejecución no puede ser mayor que el tamaño del vector binario.

8. La señalización de longitud de ejecución se puede detener cuando el número de elementos "1" señalados ha alcanzado el número máximo posible, que puede representar el tamaño máximo posible de la paleta.

[0169] Para un ejemplo con un vector de predicción binaria $\mathbf{b} = \{11001110100001\}$, el algoritmo básico descrito anteriormente produce: $[0]0-2(0)-2-3(0)-1-1(0)-4-1(1)$. Al principio, el *bin* se establece en 0, y la primera longitud de ejecución se señala a ese bin y es 0 en este ejemplo, y a continuación en cada "-" el *bin* se cambia a $1-\text{bin}$. El último indicador señalado se indica entre llaves. En los corchetes está el zero-flag. Por ejemplo, lo siguiente es el resultado de un algoritmo mejorado: $1-1(0)-1-2(0)-0-0(0)-3$.

[0170] Después de que el procesamiento del primer elemento refleje el valor de longitud de ejecución 1 debido al elemento #5, arriba, los otros valores de longitud de ejecución se reducen en 1 como se menciona en el elemento mejorado # 3, arriba. El tachado indica que no hay señalización del último indicador para el último "1" en el vector mencionado en el #2, arriba, y no hay señalización del valor de longitud de ejecución para el último elemento "1" sugerido en el #4, arriba.

[0171] Para otro ejemplo: $\mathbf{b} = \{11001110100011\}$ y un algoritmo mejorado de ejemplo da como resultado lo siguiente: $1-1(0)-1-2(0)-0-0(0)-3-1$.

[0172] Algunos procedimientos de ejemplo pueden usar un código Exponencial-Golomb para codificar la longitud de la ejecución de copia arriba, la ejecución de copia izquierda o ambas en la codificación de bloque de índice del

modo de paleta. Por ejemplo, un procedimiento que usa un código Exponencial-Golomb para codificar la longitud de la ejecución también puede aplicar un código Exponencial-Golomb para codificar la longitud de la ejecución de copia arriba, la ejecución de copia-izquierda o ambas en la codificación de bloque de índice del modo de paleta. En otro ejemplo, después de señalar los indicadores "mayor que 0", "mayor que 1" y "mayor que 2", se puede usar un código Golomb-Rice de segundo orden para codificar los valores restantes (por ejemplo, longitud de ejecución - 3) si la (por ejemplo, longitud de ejecución - 3) > 0.

[0173] En algunos ejemplos, los indicadores "mayor que 0", "mayor que 1" y "mayor que 2" pueden usarse para indicar que una longitud de ejecución es "mayor que 0", "mayor que 1" o "mayor que 2", respectivamente. Una longitud de ejecución puede codificarse usando un vector de predicción binaria codificada de longitud de ejecución.

[0174] De acuerdo con algunos ejemplos adicionales de esta divulgación, un codificador de vídeo (por ejemplo, el codificador de vídeo 20 o el descodificador de vídeo 30) puede determinar uno o más predictores de paleta nuevos. De acuerdo con estos ejemplos adicionales de esta divulgación, el codificador de vídeo puede determinar un nuevo vector booleano para señalar predictores de paleta. Estas y otras técnicas se describen con mayor detalle, por ejemplo, con respecto a la FIG. 10 a continuación.

[0175] La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de determinación de una paleta para codificar datos de vídeo, consecuente con las técnicas de esta divulgación. El ejemplo de la FIG. 10 incluye una imagen 178 que tiene una primera unidad de codificación (CU) 180 que se codifica utilizando un modo de paleta (PAL) asociado a unas primeras paletas 184 y una segunda CU 188 que está asociada a unas segundas paletas 192. Como se describe en mayor detalle a continuación y de acuerdo con las técnicas de esta divulgación, las segundas paletas 192 están basadas en las primeras paletas 184. La imagen 178 también incluye el bloque 196 codificado con un modo de codificación de predicción intra y el bloque 200 que está codificado con un modo de codificación de predicción inter.

[0176] Las técnicas de la FIG. 10 se describen en el contexto del rendimiento del codificador de vídeo 20 (FIG. 1 y FIG. 2) y el descodificador de vídeo 30 (FIG. 1 y FIG. 3) y con respecto al estándar de codificación de vídeo HEVC, con fines explicativos. Sin embargo, se debería entender que las técnicas de esta divulgación no están limitadas de esta manera, y que otros procesadores y/o dispositivos de codificación de vídeo pueden aplicarlas en otros procesos y/o estándares de codificación de vídeo.

[0177] En general, una paleta se refiere a un número de valores de píxel que son dominantes y/o representativos para una CU actualmente codificada, la CU 188 en el ejemplo de la FIG. 10. Las primeras paletas 184 y las segundas paletas 192 mostradas incluyen múltiples paletas. En algunos ejemplos, de acuerdo con aspectos de esta divulgación, un codificador de vídeo (tal como el codificador de vídeo 20 o el descodificador de vídeo 30) puede codificar paletas por separado para cada componente cromático de una CU. Por ejemplo, el codificador de vídeo 20 puede codificar una paleta para un componente de luma (Y) de una CU, otra paleta para un componente de croma (U) de la CU y otra paleta más para el componente de croma (V) de la CU. En este ejemplo, unas entradas de la paleta Y pueden representar valores Y de píxeles de la CU, unas entradas de la paleta U pueden representar valores U de píxeles de la CU y unas entradas de la paleta V pueden representar valores V de píxeles de la CU.

[0178] En otros ejemplos, el codificador de vídeo 20 puede codificar una única paleta para todos los componentes de color de una CU. En este ejemplo, el codificador de vídeo 20 puede codificar una paleta que tiene una *i*-ésima entrada que es un valor triple, que incluye Y_i , U_i y V_i . En este caso, la paleta incluye valores para cada uno de los componentes de los píxeles. En consecuencia, la representación de las paletas 184 y 192 como un conjunto de paletas que tiene múltiples paletas individuales es simplemente un ejemplo y no pretende ser limitativo.

[0179] En el ejemplo de la FIG. 10, la primera paleta 184 incluye tres entradas 202-206 que tienen un valor de índice de entrada 1, un valor de índice de entrada 2 y un valor de índice de entrada 3, respectivamente. Las entradas 202-206 relacionan los valores de índice con valores de píxel, incluyendo un valor de píxel A, un valor de píxel B y un valor de píxel C, respectivamente. Como se describe en el presente documento, en vez de codificar los valores de píxel reales de la primera CU 180, un codificador de vídeo (tal como un codificador de vídeo 20 o un descodificador de vídeo 30) puede usar la codificación basada en paleta para codificar los píxeles del bloque usando los índices 1-3. Es decir, para cada posición de píxel de la primera CU 180, el codificador de vídeo 20 puede codificar un valor de índice para el píxel, donde el valor de índice está asociado a un valor de píxel en una o más de las primeras paletas 184. El descodificador de vídeo 30 puede obtener los valores de índice a partir de una corriente de bits y reconstruir los valores de píxel usando los valores de índice y una o más de las primeras paletas 184. Por tanto, el codificador de vídeo 20 transmite unas primeras paletas 184 en una corriente de bits de datos de vídeo codificada para su uso por el descodificador de vídeo 30 en la descodificación basada en paleta.

[0180] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden determinar segundas paletas 192 basándose en las primeras paletas 184. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden ubicar uno o más bloques a partir de los cuales se determinan las paletas predictivas, en este ejemplo, las primeras paletas 184. En algunos ejemplos, tales como el ejemplo ilustrado en la

FIG. 10, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden localizar la CU previamente codificada como una CU izquierda contigua (primera CU 180) al determinar una paleta predictiva para la segunda CU 188.

[0181] En el ejemplo de la FIG. 10, las segundas paletas 192 incluyen tres entradas 208-212 que tienen un valor de índice de entrada 1, un valor de índice de entrada 2 y un valor de índice de entrada 3, respectivamente. Las entradas 208 a 212 relacionan los valores de índice con los valores de píxel, incluyendo el valor de píxel A, el valor de píxel B y el valor de píxel D, respectivamente. En este ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos que indican qué entradas de las primeras paletas 184 están incluidas en las segundas paletas 192. En el ejemplo de la FIG. 10, el uno o más elementos sintácticos se ilustran como el vector 216 (por ejemplo, un vector de predicción binaria). El vector 216 tiene una serie de bins (o bits) asociados, indicando cada bin si el predictor de paleta asociado a ese bin se usa para predecir una entrada de la paleta actual. Por ejemplo, el vector 216 indica que las dos primeras entradas de las primeras paletas 184 (202 y 204) están incluidas en las segundas paletas 192 (un valor de "1" en el vector 216), mientras que la tercera entrada de las primeras paletas 184 no está incluida en las segundas paletas 192 (un valor de "0" en el vector 216). En el ejemplo de la FIG. 10, el vector es un vector booleano.

[0182] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden determinar una lista de predictores de paleta (que también puede denominarse tabla de predictores de paleta) cuando se realiza la predicción de paleta. La lista de predictores de paleta puede incluir entradas de paletas de uno o más bloques contiguos que se usan para predecir una o más entradas de una paleta para codificar un bloque actual. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden construir la lista de la misma manera. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar datos (tales como el vector 216) para indicar qué entradas de la lista de predictores de paleta se tienen que incluir en una paleta para codificar un bloque actual.

[0183] El documento C. Gisquet, G. Laroche, y P. Onno, "AhG10: Palette predictor stuffing", JCTVC-Q0063 divulga un proceso de ejemplo para determinar una lista de predictores de paleta. En algunos ejemplos, como se indicó anteriormente, el codificador de vídeo 20 o el descodificador de vídeo 30 puede usar un vector booleano (como el vector 216) para indicar si cada elemento en la 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.

[0184] En algunos ejemplos, todos los elementos de la lista de predictores de paleta se obtienen de la paleta previamente codificada (por ejemplo, la paleta codificada con el bloque previamente codificado). Sin embargo, tales paletas pueden estar espacialmente alejadas de la CU actual, lo cual puede hacer que la correlación de la paleta sea relativamente débil. En general, la expansión de la tabla (o lista) de predictores de paleta puede ser útil (por ejemplo, puede proporcionar predictores más precisos, lo cual puede dar como resultado una ganancia de eficiencia). Sin embargo, la determinación y el uso de una tabla (o lista) de predictores de paleta relativamente grande da como resultado un vector booleano relativamente más largo.

[0185] En varios ejemplos, las técnicas descritas en esta divulgación pueden incluir técnicas para diversas combinaciones de paletas de determinación, predicción y/o señalización en la codificación basada en paleta. Las técnicas pueden incluir cualquier combinación de determinar predictores de paleta, agregar predictores a una lista de predictores candidatos, eliminar predictores de una lista de predictores candidatos, codificar una indicación del uso de predictores candidatos o cualquier otra técnica descrita en el presente documento. Si bien ciertos ejemplos pueden describirse individualmente con fines ilustrativos y de claridad, esta divulgación contempla cualquier combinación de las técnicas para la codificación basada en paleta descritas en el presente documento.

[0186] De acuerdo con algunos aspectos de esta divulgación, un codificador de vídeo (tal como el codificador de vídeo 20 o el descodificador de vídeo 30) puede determinar uno o más nuevos predictores de paleta. Por ejemplo, ciertas técnicas de esta divulgación incluyen la determinación de uno o más predictores de paleta espacial. En un ejemplo, si la CU por encima de la CU actual (denominada "CU por encima" o "CU superior" y mostrada como CU 196 intracodificada en el ejemplo de la FIG. 10) usa codificación basada en paleta, la paleta de la CU superior es efectiva como un predictor de paleta para la CU actual (segunda CU 188). Es decir, cada una de las entradas de la CU superior puede incluirse en una lista de predictores de paleta (por ejemplo, que puede estar asociada con un vector que indica qué entradas se usan como predictores) para predecir una paleta actual.

[0187] Sin embargo, si la CU anterior es parte de una CTU diferente, el uso de la paleta de la CU anterior para la predicción puede requerir almacenamiento adicional. Por ejemplo, para un bloque entre predicciones, el codificador de vídeo puede tener que acceder a la memoria (por ejemplo, una memoria externa al codificador de vídeo), para reconstruir los valores de píxeles, lo cual puede dar como resultado latencia. Sin embargo, el codificador de vídeo puede, en algunos ejemplos, considerar las muestras reconstruidas asociadas con un bloque con predicción inter, por el presente denotado como muestras de referencia. Dichas muestras reconstruidas pueden almacenarse localmente y estar disponibles tanto para el codificador de vídeo 20 como para el descodificador de vídeo 30.

[0188] En un ejemplo, de acuerdo con los aspectos de esta divulgación, las muestras de referencia se agregan a la lista de predictores de paleta. Por ejemplo, los valores de píxeles de un bloque espacialmente adyacente (que

5 puede estar codificado en paleta o intracodificado) pueden identificarse como predictores de paleta candidatos y asociarse con un vector (como se indicó anteriormente). En algunos ejemplos, solo se puede usar un subconjunto de las muestras de referencia. Por ejemplo, si la lista de predictores de paleta ya incluye la paleta de una CU particular, como la CU contigua izquierda (primera CU 180), las muestras de esa CU pueden no estar incluidas en la lista de predicción de paleta. Además, en algunos ejemplos, el codificador de vídeo puede aplicar un proceso de eliminación para eliminar predictores de paleta duplicados en la lista, a costa de una complejidad adicional para el codificador de vídeo 20 y especialmente el descodificador de vídeo 30. Estos nuevos predictores de paleta se pueden insertar al principio de la lista de predictores o al final de la lista. En otros ejemplos, los nuevos predictores de paleta (por ejemplo, Predictores de paleta espacial) pueden insertarse de forma adaptativa en la lista de acuerdo con una determinada regla, como el tamaño de la lista de predictores, el tamaño de CU o similares.

[0189] En otros ejemplos, cualquier otro píxel reconstruido, como los píxeles en cualquier CU en una columna a la izquierda de la CU actual, también se puede agregar a la lista de predictores de paleta.

15 [0190] De esta manera, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar uno o más predictores de paleta espacial para predecir una o más entradas de una paleta para un bloque actual de datos de vídeo, donde cada entrada de una o más entradas indica un valor de píxel, y codificar uno o más elementos sintácticos que indiquen si cada predictor de paleta respectivo del uno o más predictores de paleta se usa para predecir entradas de la paleta para el bloque actual de datos de vídeo.

20 [0191] De acuerdo con otros aspectos de esta divulgación, un codificador de vídeo (tal como el codificador de vídeo 20 o el descodificador de vídeo 30) puede determinar un nuevo vector booleano para señalar predictores de paleta. En algunos ejemplos, las técnicas de esta divulgación pueden usarse para codificar el vector booleano para la predicción de la paleta.

25 [0192] Por ejemplo, de acuerdo con los aspectos de esta divulgación, el codificador de vídeo 20 o el descodificador de vídeo 30 pueden determinar un árbol binario para la predicción de la paleta. En un ejemplo con fines ilustrativos, deje que el vector booleano 0-1 que representa el uso de predictores de paleta sea denotado por N. Para codificar eficientemente este vector, se usa un árbol binario. Específicamente, el vector N se divide en varias regiones contiguas, con cada región que contiene uno o más elementos de un vector. Se señala un bit para indicar si todos los elementos dentro de esa región son cero o no. Si la región no es toda cero, se divide en subregiones de la misma manera.

30 [0193] La partición puede detenerse de acuerdo con las reglas dadas. Por ejemplo, una regla puede ser "si el tamaño de la región es menor que un umbral, el proceso de partición se detiene". Cuando se detiene la partición, si el tamaño de la región es mayor que uno y contiene elementos distintos de cero, cada elemento de la región se señala en el flujo de bits. En este caso, suponiendo que el tamaño de la región es X, si los primeros elementos X-1 son todos cero, el último elemento debe ser 1, de modo que no pueda incluirse en el flujo de bits.

35 [0194] En otro ejemplo, cuando el proceso de partición se detiene y el tamaño de la región es mayor que uno pero menor que un segundo umbral, el codificador de vídeo puede obtener la señalización de un bit para indicar si todos los elementos en esa región son cero. En cambio, para cada elemento en la región, el codificador de vídeo puede señalar un bit para indicar si el elemento respectivo es cero o 1.

40 [0195] En otro ejemplo, en lugar de un árbol binario, se puede usar un árbol con más de dos ramas en cada nodo. Además, en algunos ejemplos, se puede obtener la codificación de los primeros niveles a partir de la raíz del árbol. En tales ejemplos, se puede suponer implícitamente que los valores son 1.

45 [0196] De acuerdo con los aspectos de esta divulgación, la posición de la última entrada distinta de cero de la lista de predictores de paleta (es decir, el último 1 en el vector booleano) puede codificarse explícitamente en el flujo de bits, antes de indicar si las entradas individuales son 0 o 1. En este caso, los elementos desde la última posición (inclusive) hasta el final del vector booleano no se señalan en el flujo de bits. En otro ejemplo, la posición del primer predictor de paleta que se usa para la predicción, es decir, el primer 1 en el vector booleano, también puede indicarse explícitamente en el flujo de bits.

50 [0197] De acuerdo con los aspectos de esta divulgación, el número de entradas que se utilizan de la lista de predictores de paleta puede codificarse explícitamente en el flujo de bits. Por ejemplo, las técnicas de esta divulgación incluyen la señalización de ese número de unos ("1") en el vector booleano analizado anteriormente. En algunos ejemplos, el codificador de vídeo 20 puede indicar inicialmente el número de "1s" (*TotalOnes*) en un vector booleano. A continuación, el codificador de vídeo 20 puede señalar secuencialmente (o señalar de alguna otra manera) cada elemento en el vector booleano (0 o 1) hasta que el número de 1 señalados sea igual a *TotalOnes*.

55 [0198] Asimismo, el descodificador de vídeo 30 puede descodificar primero una indicación del número de los que están incluidos en un vector booleano recibido. A continuación, el descodificador de vídeo 30 puede descodificar cada elemento del vector recibido hasta que el descodificador de vídeo 30 haya descodificado el número de "Es"

en el vector que coincida con el número de "Es" indicado. En algunos ejemplos, la señalización por elementos (por ejemplo, para los elementos del vector) puede ser secuencial desde el comienzo del vector. En otros ejemplos, los elementos del vector pueden señalarse de acuerdo con algún otro orden de escaneo. En otros ejemplos, la señalización por elementos puede usar tecnología de codificación de longitud de ejecución.

[0199] En algunos ejemplos, cuando el número de unidades en el vector booleano se señala inicialmente (por ejemplo, *TotalOnes* se señala en el flujo de bits antes del vector booleano), cada elemento sintáctico (por ejemplo, indicador) en el vector booleano puede codificarse en contexto usando CABAC. En algunos casos, se pueden usar múltiples contextos. En algunos ejemplos, todos los elementos sintácticos del vector booleano pueden estar codificados por contexto. En otros ejemplos, solo un subconjunto de los elementos sintácticos del vector booleano puede codificarse según el contexto.

[0200] De acuerdo con los aspectos de esta divulgación, el contexto para un elemento sintáctico del vector booleano puede depender de la posición del elemento sintáctico (por ejemplo, un indicador binario) dentro del vector booleano y el número de unos *señalados* (*TotalOnes*) que son inicialmente señalado. En un ejemplo con fines ilustrativos, si $((\text{position} > 3) \ \&\& \ ((\text{position}/2 + 1) > \text{TotalOnes}))$ un codificador de vídeo (como el codificador de vídeo 20 o el descodificador de vídeo 30) puede usar un primer contexto; de lo contrario (cuando la condición no es verdadera), el codificador de vídeo puede usar otro contexto, donde la *position* indica una posición relativa del elemento sintáctico que se codifica en el vector booleano. En este caso, el codificador de vídeo utiliza dos contextos para codificar los elementos sintácticos (por ejemplo, indicadores binarios) en el vector booleano, y selecciona un contexto para un elemento sintáctico particular del vector booleano basándose en la posición relativa del elemento sintáctico particular en el vector booleano. En otro ejemplo, el codificador de vídeo puede usar un contexto si una condición $((\text{position} > 3) \ \&\& \ ((\text{position}/4 + 1) > = \text{TotalOnes}))$ es verdadera y otro contexto si la condición no es verdadera. También se pueden usar otras condiciones para determinar contextos.

[0201] En los ejemplos anteriores, el contexto CABAC para un elemento sintáctico particular de un vector booleano depende de la posición relativa del elemento sintáctico y del número total de unos en el vector booleano (*TotalOnes*). En otros ejemplos, en lugar de depender de la posición y *TotalOnes*, el codificador de vídeo puede determinar el contexto para los elementos sintácticos basándose en una posición relativa del elemento sintáctico que se está codificando y el número total de elementos que se han señalado hasta ahora (por ejemplo, el número total de los que han ocurrido antes de que el elemento sintáctico se codifique en el vector booleano). En otros ejemplos, un codificador de vídeo puede usar cualquier combinación de las condiciones descritas anteriormente para determinar el contexto de los elementos sintácticos de codificación CABAC de un vector booleano.

[0202] De acuerdo con aspectos de esta divulgación, un codificador señala el tamaño de una paleta y un descodificador lo recibe. Por ejemplo, se pueden indicar varias entradas en una paleta para indicar el tamaño de la paleta. En algunos ejemplos, el codificador de vídeo 20 puede indicar inicialmente el tamaño de la paleta (*PLTSize*). A continuación, el codificador de vídeo 20 puede señalar cada elemento en el vector booleano (0 o 1) hasta que el número de 1 señalados sea igual a *PLTSize*. El descodificador de vídeo 30 puede recibir una indicación del tamaño de la paleta, y puede continuar agregándose a la paleta (por ejemplo, usando predictores de paleta identificados por un vector) hasta que el número de entradas en la paleta alcance el tamaño de paleta señalado.

[0203] En algunos ejemplos, el codificador de vídeo 20 puede indicar inicialmente un tamaño de paleta (*PLTSize*). A continuación, el codificador de vídeo 20 puede indicar el número de elementos de paleta no predichos (*nonPredPLTSize*). A continuación, el codificador de vídeo 20 puede señalar cada elemento en el vector booleano (0 o 1) hasta que el número de "1s" señalado sea igual a una diferencia entre el tamaño de la paleta y el número de elementos de paleta no predichos ($\text{PLTSize} - \text{nonPredPLTSize}$). En algunos ejemplos, el descodificador de vídeo 30 puede recibir una indicación del tamaño de la paleta y el número de entradas en la paleta que no se predicen usando un predictor de paleta (por ejemplo, el número de "0" de un vector). El descodificador de vídeo 30 también puede recibir un vector que indica qué predictores de paleta se incluirán en una paleta (por ejemplo, como el vector booleano descrito anteriormente). El descodificador de vídeo 30 puede descodificar cada elemento en el vector hasta que el número de "Es" sea igual a una diferencia entre el tamaño de la paleta y el número de elementos de paleta no predichos ($\text{PLTSize} - \text{nonPredPLTSize}$).

[0204] De esta manera, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden, en un ejemplo, determinar uno o más predictores de paleta para predecir una o más entradas de una paleta para un bloque actual de datos de vídeo, en el que cada entrada de una o más entradas indican un valor de píxel, determinar un vector que tenga una pluralidad de ubicaciones, en el que cada ubicación del vector indica si se usa un predictor de paleta respectivo de uno o más predictores de paleta para predecir una entrada de la paleta para el bloque actual de datos de vídeo, dividir el vector en una o más regiones y determinar uno o más valores de los bins del vector basándose en la partición.

[0205] Si bien las técnicas de la FIG. 10 se han descrito anteriormente en el contexto de las CU (HEVC), debería entenderse que las técnicas también pueden aplicarse a unidades de predicción (PU) o en otros procesos y/o normas de codificación de vídeo.

[0206] Se debe reconocer que, dependiendo del ejemplo, determinados actos o acontecimientos de cualquiera de las técnicas descritas en el presente documento se pueden realizar en una secuencia distinta, se pueden añadir, combinar o excluir por completo (por ejemplo, no todos los actos o acontecimientos descritos son necesarios para la puesta en práctica de las técnicas). Por otro lado, en determinados ejemplos, los actos o acontecimientos pueden tener lugar simultáneamente, por ejemplo, a través de procesamiento de múltiples subprocesos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente. Además, aunque con propósitos de claridad se describe que un único módulo o unidad realiza determinados aspectos de la presente divulgación, se debería entender que una combinación de unidades o módulos asociados a un codificador de vídeo pueden realizar las técnicas de la presente divulgación.

[0207] Determinados aspectos de esta divulgación se han descrito con respecto al estándar de HEVC en desarrollo, con propósitos ilustrativos. Sin embargo, las técnicas descritas en esta divulgación pueden ser útiles para otros procesos de codificación de vídeo, que incluyen otros procesos de codificación de vídeo, estándar o patentados, aún no desarrollados.

[0208] El codificador de vídeo 20 (FIGS. 1 y 2) y/o el decodificador de vídeo 30 (FIGS. 1 y 3), ambos de los cuales se pueden denominar en general codificadores de vídeo, pueden realizar las técnicas descritas anteriormente. Del mismo modo, la codificación de vídeo se puede referir a una codificación de vídeo o una decodificación de vídeo, según corresponda.

[0209] Las técnicas descritas en esta divulgación pueden incluir técnicas para diversas combinaciones de uno o más aspectos diferentes de la codificación de paleta.

[0210] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, un programa informático, firmware o cualquier combinación de los mismos. Si se implementan en un programa informático, las funciones se pueden almacenar en o transmitirse por un medio legible por ordenador como una o más instrucciones o códigos y ejecutar mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que son no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0211] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria *flash* o cualquier otro medio que se pueda usar para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. También, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debería entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se dirigen a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, de los cuales el disco flexible habitualmente reproduce datos magnéticamente, mientras que los demás discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior se deberían incluir también dentro del alcance de los medios legibles por ordenador.

[0212] 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), arreglos de compuertas programables en el campo (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquier estructura anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar en módulos de hardware y/o programa informático dedicados configurados para codificar y decodificar, o incorporados en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

- 5 **[0213]** Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores, como se ha descrito anteriormente, conjuntamente con un programa informático y/o firmware adecuados.
- 10 **[0214]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones. Los procedimientos anteriores se pueden usar individualmente o en combinación.

REIVINDICACIONES

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

5 recibir (900) un vector de predicción binaria codificada para un bloque actual de datos de vídeo;

descodificar (902) el vector de predicción binaria codificada usando una técnica de descodificación de longitud de ejecución, en el que la técnica de descodificación de longitud de ejecución comprende:

10 descodificar el vector de predicción binaria codificada para obtener valores de longitud de ejecución descodificados;

para cada valor de longitud de ejecución descodificado:

15 si el valor de longitud de ejecución descodificado es mayor que un valor de longitud de ejecución reservado L, restar 1 del valor de longitud de ejecución descodificado para dar un valor de longitud de ejecución real que especifique un número de ceros consecutivos que preceden a un valor distinto de cero,

20 si el valor de longitud de ejecución descodificado es menor que el valor de longitud de ejecución reservado L, utilizar el valor de longitud de ejecución descodificado como el valor de longitud de ejecución real, o

si el valor de longitud de ejecución descodificado es igual al valor de longitud de ejecución reservado L, determinar que no hay más valores distintos de cero en el vector de predicción binaria;

25 generar (904) una paleta para el bloque actual de datos de vídeo basándose en el vector de predicción binaria, con el vector de predicción binaria que comprende entradas que indican si las entradas de paleta usadas anteriormente se reutilizan o no para la paleta para el bloque actual de datos de vídeo; y

30 descodificar (906) el bloque de datos de vídeo actual usando la paleta.

2. El procedimiento según la reivindicación 1, en el que generar la paleta para el bloque actual de datos de vídeo basándose en el vector de predicción binaria comprende:

35 copiar, desde una memoria intermedia, entradas de paleta usadas previamente en la paleta que se indican como reutilizadas para la paleta por el vector de predicción binaria; y

40 recibir, en el caso de que el número de entradas de paleta utilizadas previamente copiadas para la paleta sea menor que un tamaño máximo de paleta, entradas de paleta adicionales.

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

45 generar (800) una paleta para el bloque actual de datos de vídeo;

generar (802) un vector de predicción binaria para la paleta para el bloque actual de datos de vídeo, con el vector de predicción binaria comprende entradas que indican si las entradas de paleta utilizadas anteriormente se reutilizan o no para la paleta para el bloque actual de datos de vídeo;

50 codificar (804) el vector de predicción binaria usando una técnica de codificación de longitud de ejecución, en el que la técnica de codificación de longitud de ejecución comprende:

para cada valor distinto de cero en el vector de predicción binaria:

55 si un número de ceros consecutivos que preceden al valor distinto de cero es igual o mayor que un valor de longitud de ejecución reservado L, sumar 1 al número de ceros consecutivos y codificar el resultado como un valor de longitud de ejecución, o

60 si el número de ceros consecutivos que preceden al valor distinto de cero es menor que el valor de longitud de ejecución reservado L, codificar el número de ceros consecutivos como el valor de longitud de ejecución; y

65 para varios ceros consecutivos que siguen a un valor final distinto de cero en el vector de predicción binaria, codificar el valor de longitud de ejecución reservado L como el valor de longitud de ejecución; y

codificar (806) el bloque actual de datos de vídeo usando la paleta.

- 5 **4.** El procedimiento según la reivindicación 1 o la reivindicación 3, donde en el valor de longitud de ejecución reservado L es 1.
- 5.** El procedimiento según la reivindicación 2 o la reivindicación 3, que comprende además:
- 10 almacenar las entradas de paleta utilizadas anteriormente para uno o más bloques de datos de vídeo previamente codificados en la memoria intermedia.
- 6.** El procedimiento según la reivindicación 5, que comprende además:
- 15 eliminar entradas duplicadas de las entradas de paleta usadas previamente almacenadas en la memoria intermedia.
- 7.** El procedimiento según la reivindicación 5, en el que las entradas de paleta usadas previamente para uno o más bloques de datos de vídeo previamente codificados comprenden entradas de paleta usadas previamente para una línea de píxeles sobre el bloque actual de datos de vídeo y entradas de paleta usadas previamente para una línea de píxeles a la izquierda del bloque actual de datos de vídeo.
- 20 **8.** Un aparato configurado para descodificar datos de vídeo, comprendiendo el aparato:
- medios para recibir un vector de predicción binaria codificada para un bloque actual de datos de vídeo;
- 25 medios para descodificar el vector de predicción binaria codificada utilizando una técnica de descodificación de longitud de ejecución, en el que los medios para descodificar el vector de predicción binaria codificada comprenden medios para descodificar el vector de predicción binaria codificada para obtener valores de longitud de ejecución descodificados de manera que, para cada valor de longitud de ejecución descodificado:
- 30 si el valor de longitud de ejecución descodificado es mayor que un valor de longitud de ejecución reservado L, restar 1 del valor de longitud de ejecución descodificado para dar un valor de longitud de ejecución real, lo cual especifica un número de ceros consecutivos que preceden a un valor distinto de cero,
- 35 si el valor de longitud de ejecución descodificado es menor que el valor de longitud de ejecución reservado L, utilizar el valor de longitud de ejecución descodificado como el valor de longitud de ejecución real, o
- 40 si el valor de longitud de ejecución descodificado es igual al valor de longitud de ejecución reservado L, determinar que no hay más valores distintos de cero en el vector de predicción binaria; medios para generar una paleta para el bloque actual de datos de vídeo basado en el vector de predicción binaria, con el vector de predicción binaria que comprende entradas que indican si las entradas de paleta usadas previamente se reutilizan o no para la paleta para el bloque actual de datos de vídeo; y
- 45 medios para descodificar el bloque actual de datos de vídeo usando la paleta.
- 9.** Un aparato configurado para codificar datos de vídeo, comprendiendo el aparato:
- 50 medios para generar una paleta para el bloque actual de datos de vídeo;
- medios para generar un vector de predicción binaria para la paleta para el bloque actual de datos de vídeo, con el vector de predicción binaria que comprende entradas que indican si las entradas de paleta usadas previamente se reutilizan o no para la paleta para el bloque actual de datos de vídeo;
- 55 medios para codificar el vector de predicción binaria usando una técnica de codificación de longitud de ejecución, en el que la técnica de codificación de longitud de ejecución comprende:
- 60 para cada valor distinto de cero en el vector de predicción binaria:
- si un número de ceros consecutivos que preceden al valor distinto de cero es igual o mayor que un valor de longitud de ejecución reservado L, sumar 1 al número de ceros consecutivos y codificar el resultado como un valor de longitud de ejecución, o
- 65

si el número de ceros consecutivos que preceden al valor distinto de cero es menor que el valor de longitud de ejecución reservado L, codificar el número de ceros consecutivos como el valor de longitud de ejecución; y

5 para varios ceros consecutivos que siguen a un valor final distinto de cero en el vector de predicción binaria, codificar el valor de longitud de ejecución reservado L como el valor de longitud de ejecución; y

10 medios para codificar el bloque actual de datos de vídeo usando la paleta.

10. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo configurados para descodificar datos de vídeo realicen el procedimiento de acuerdo con cualquiera de las reivindicaciones anteriores 1 a 7.

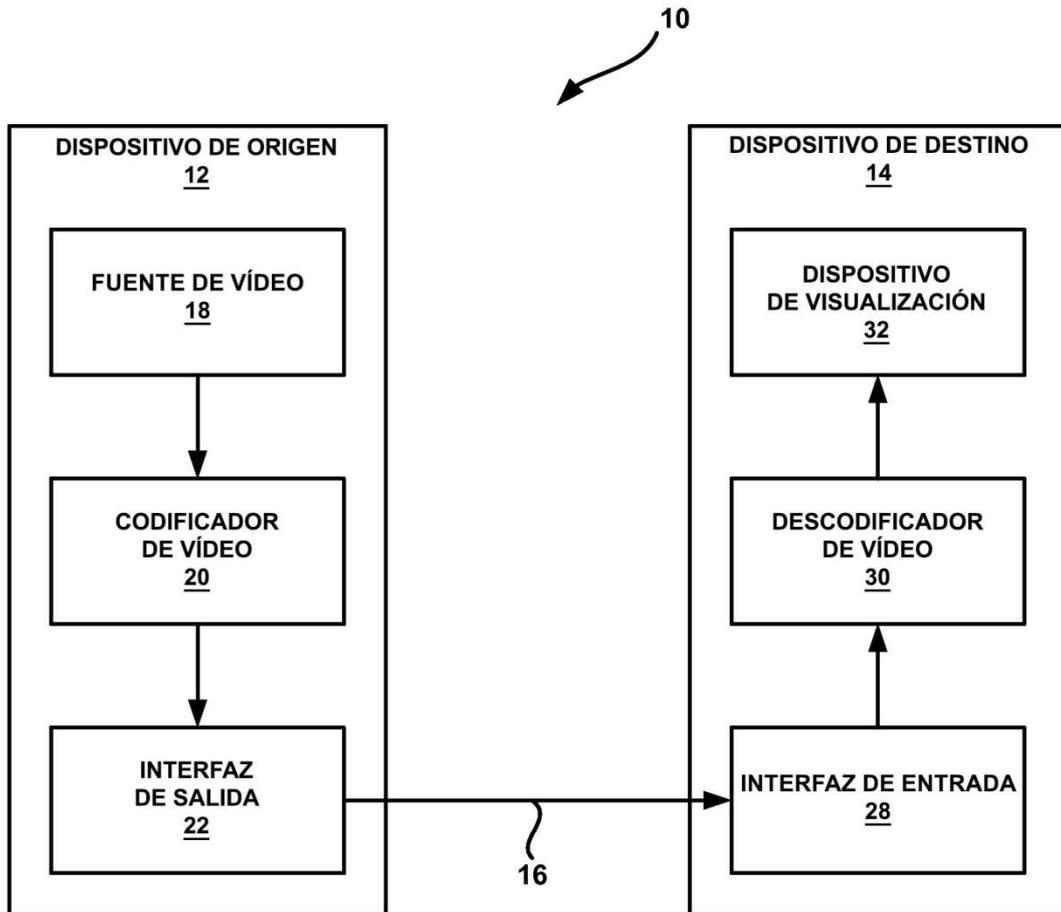


FIG. 1

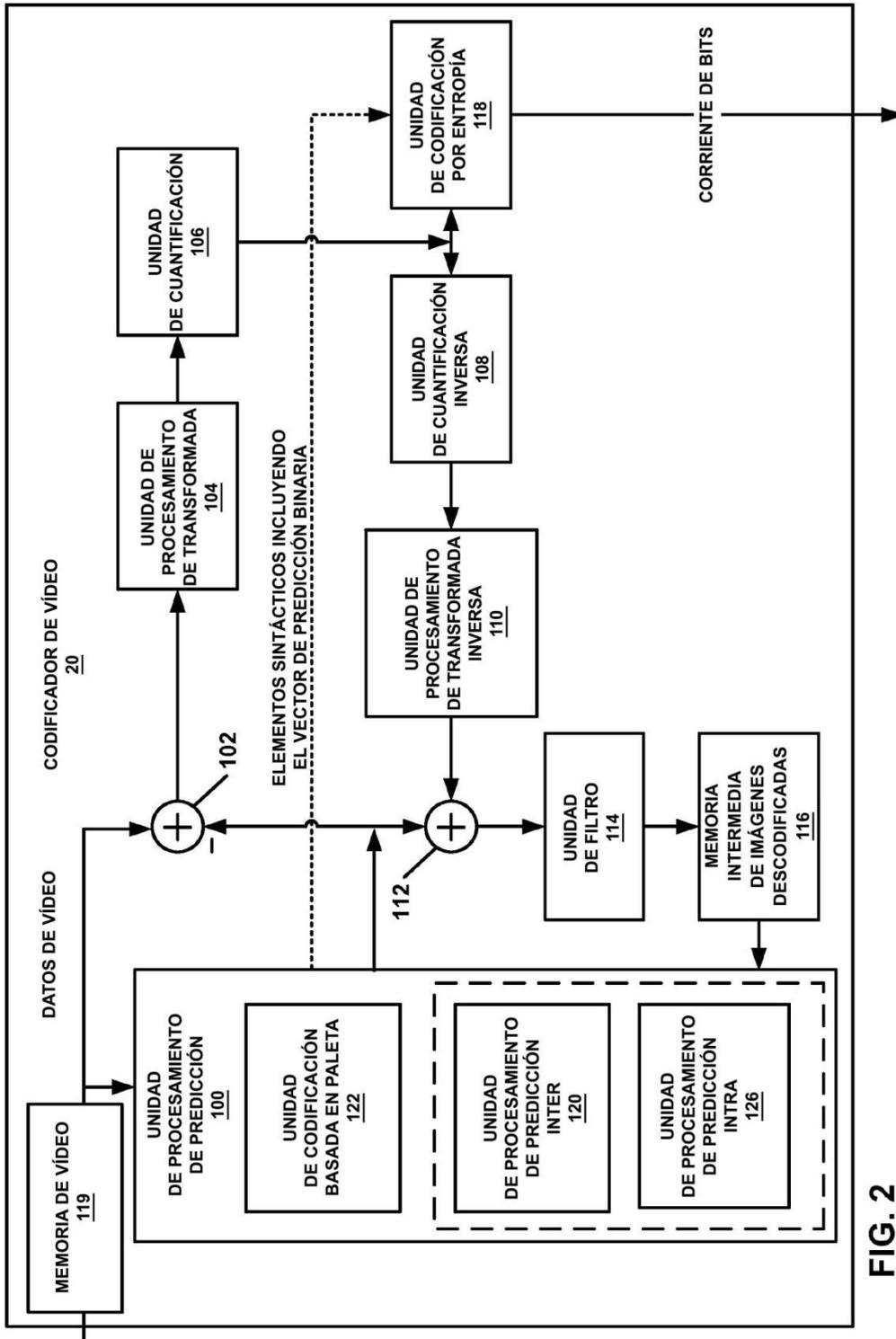


FIG. 2

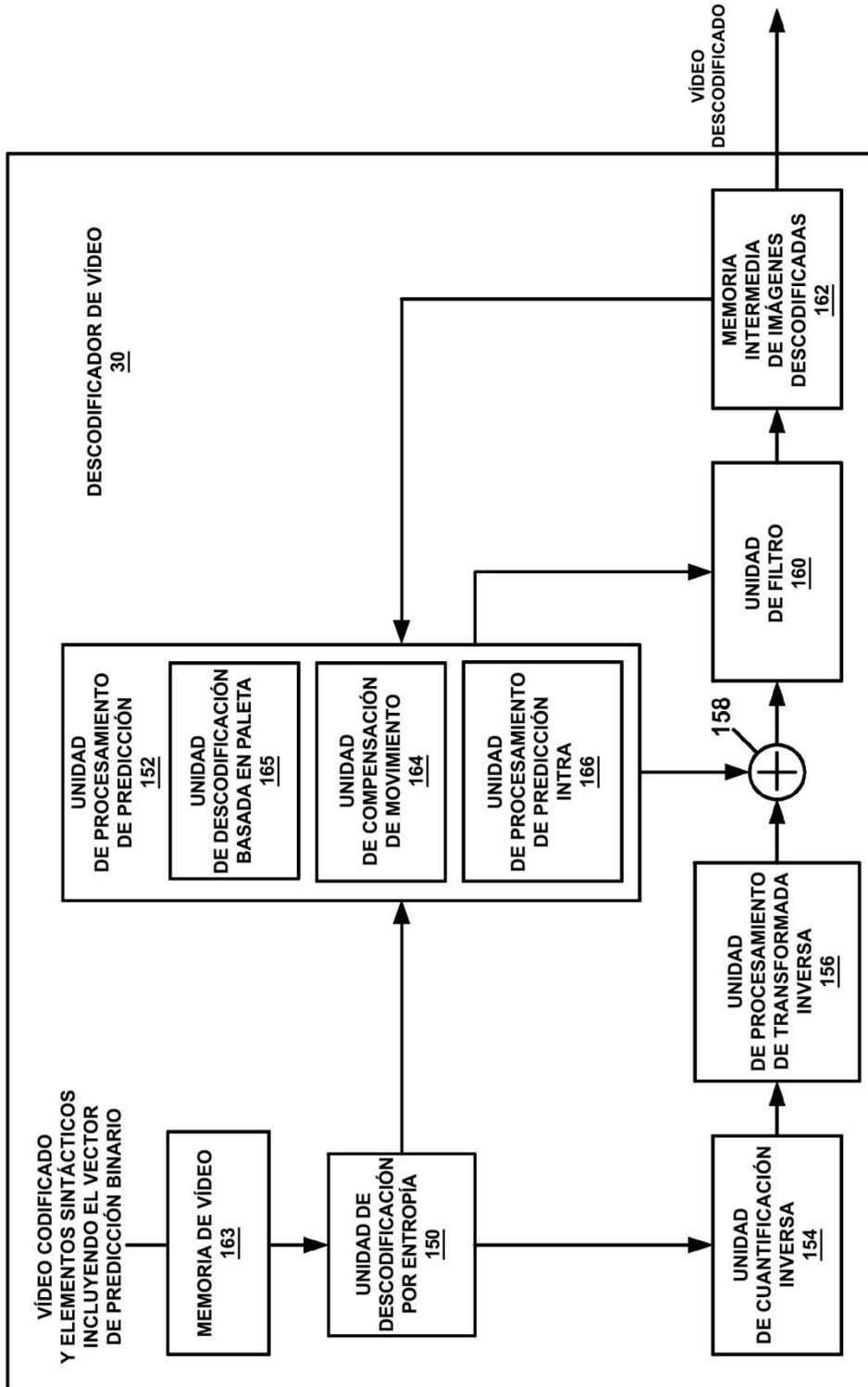


FIG. 3

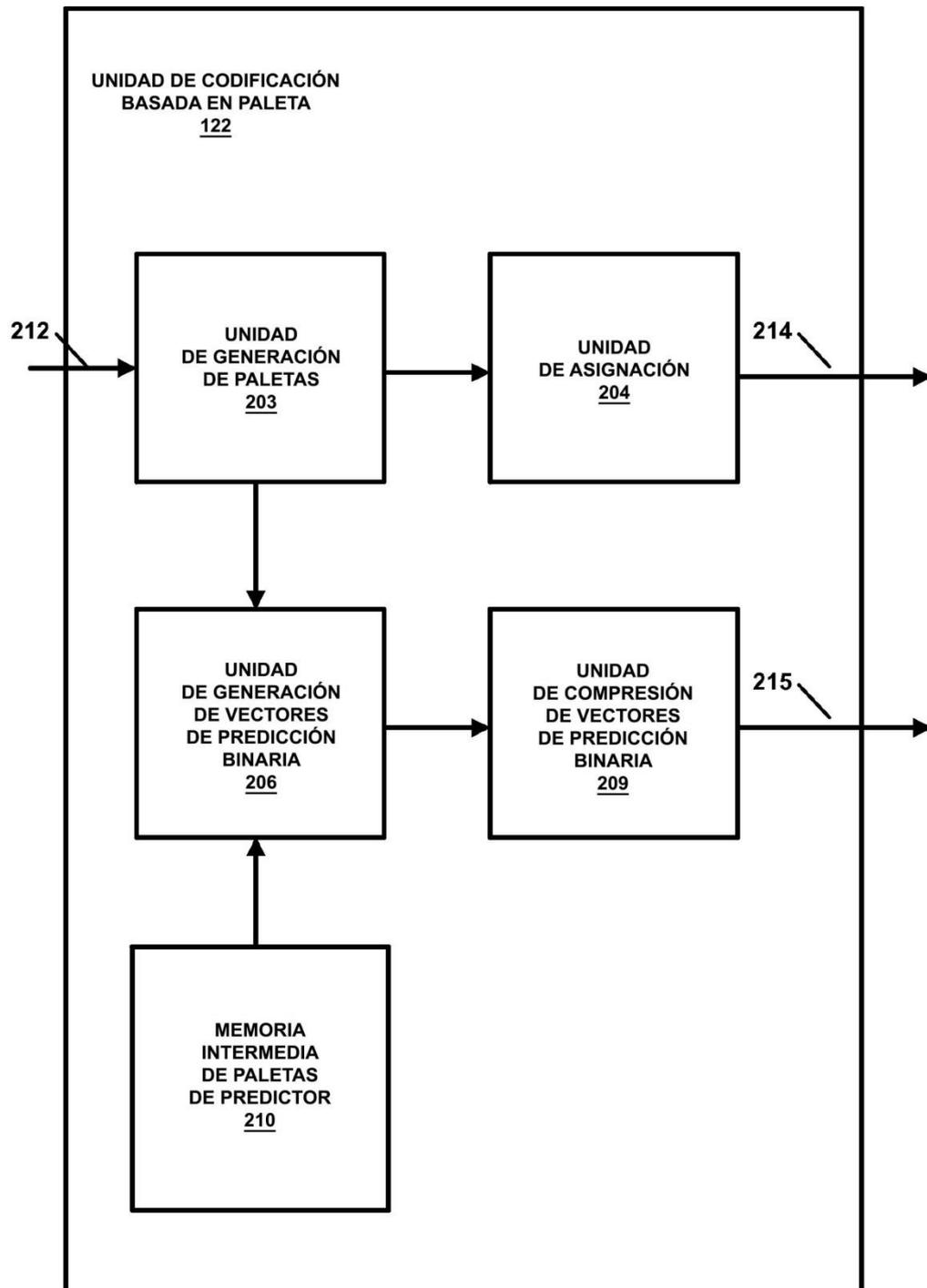


FIG. 4

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

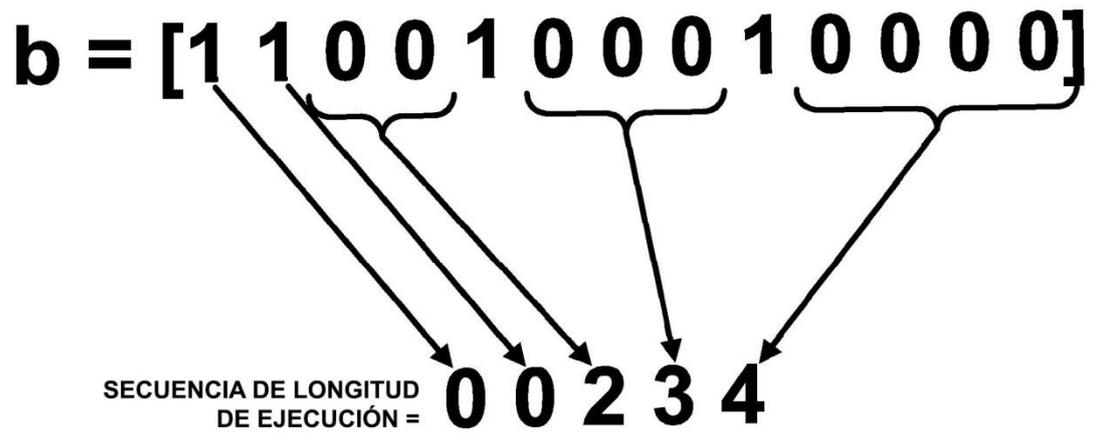


FIG. 6

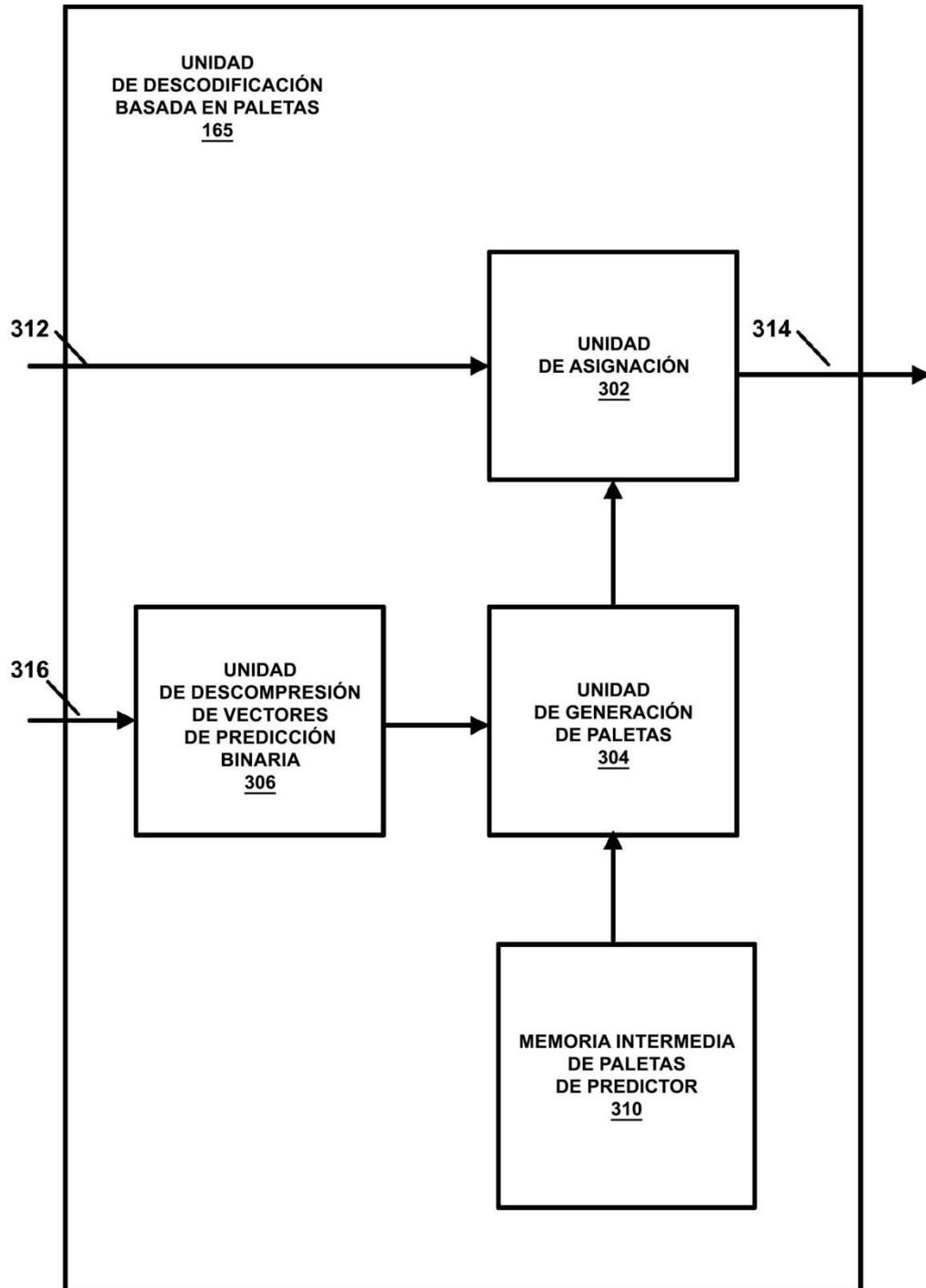


FIG. 7

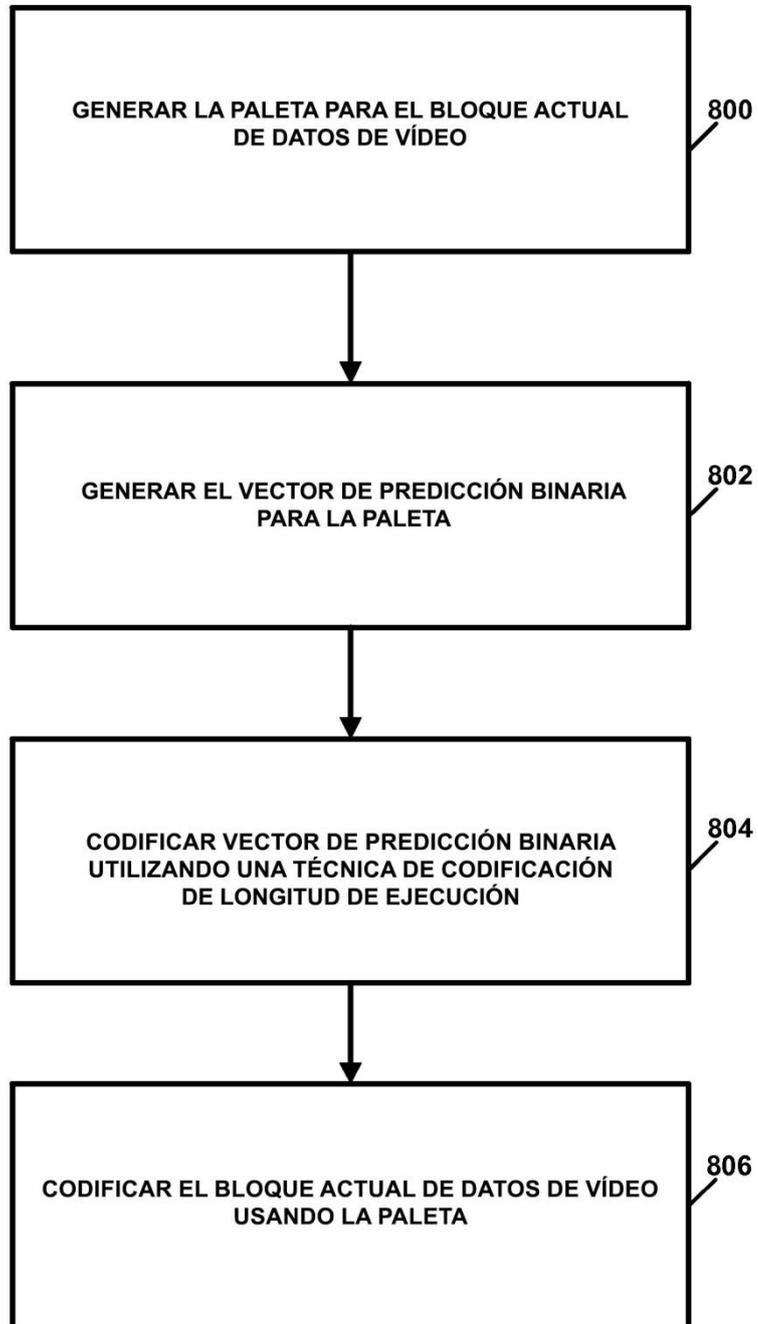


FIG. 8

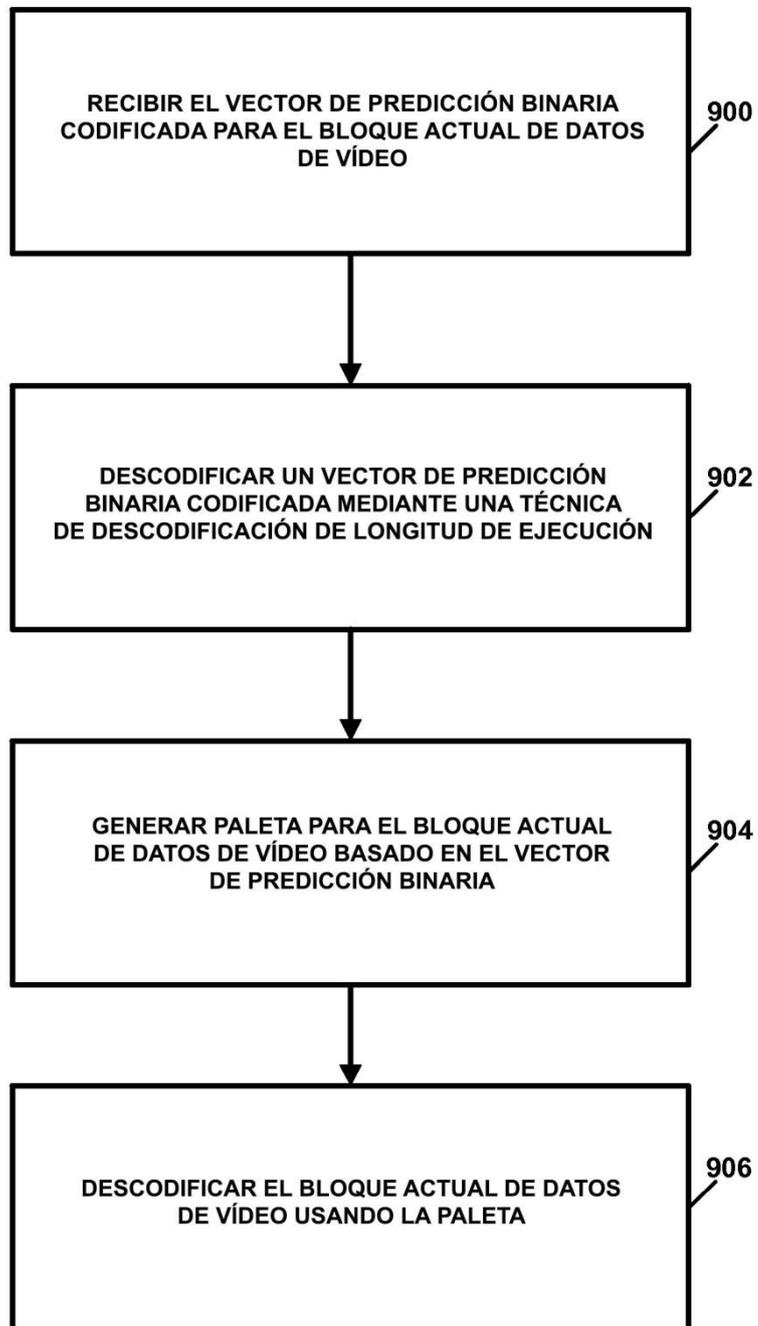


FIG. 9

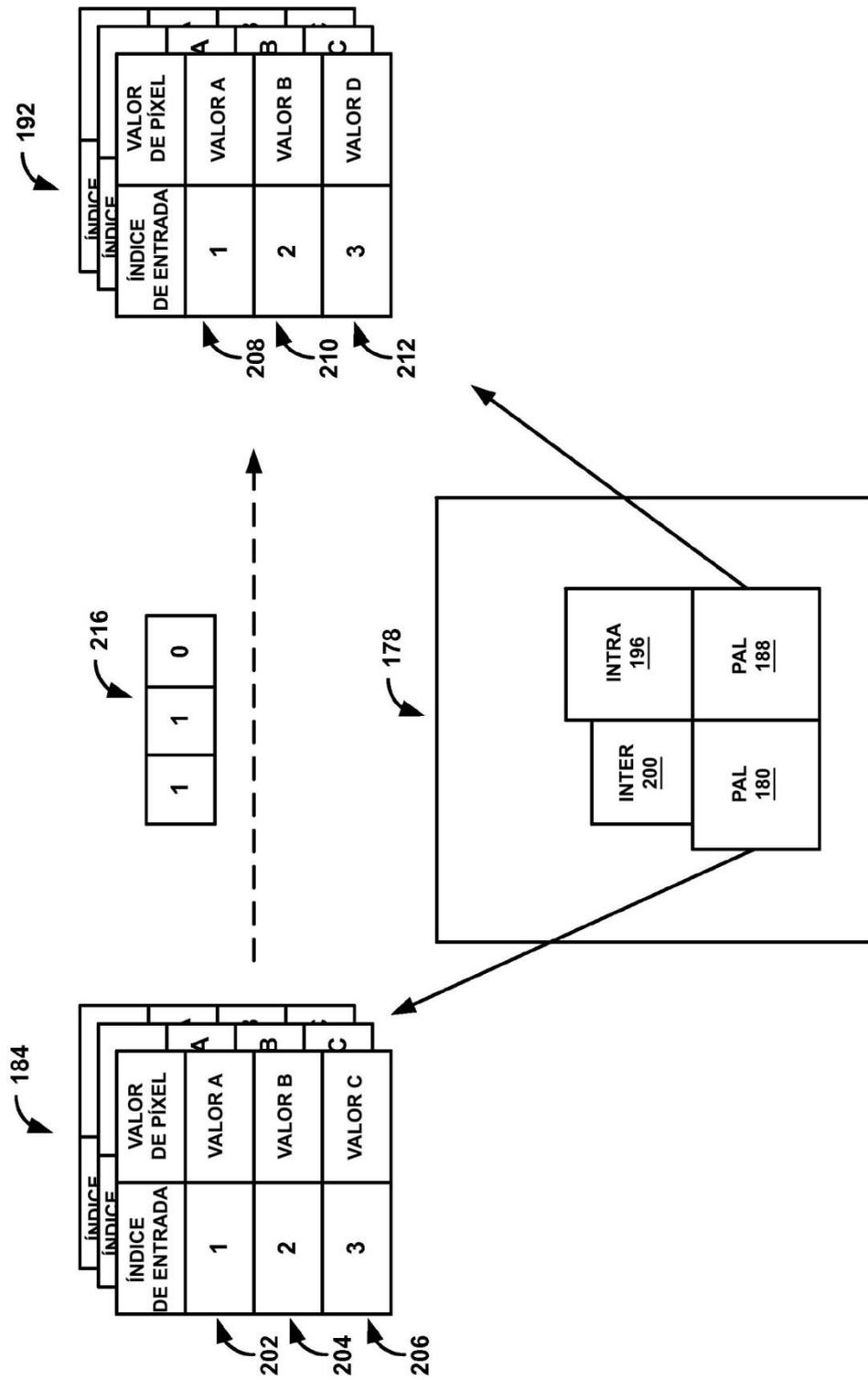


FIG. 10