

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 633 947**

51 Int. Cl.:

H04N 19/70 (2014.01)
H04N 19/196 (2014.01)
H04N 19/593 (2014.01)
H04N 19/93 (2014.01)
H04N 19/94 (2014.01)
G09G 5/06 (2006.01)
G06T 9/00 (2006.01)
H04N 1/64 (2006.01)
H04N 19/176 (2014.01)
H04N 19/186 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **11.07.2014 PCT/US2014/046402**
- 87 Fecha y número de publicación internacional: **15.01.2015 WO15006724**
- 96 Fecha de presentación y número de la solicitud europea: **11.07.2014 E 14744735 (3)**
- 97 Fecha y número de publicación de la concesión europea: **26.04.2017 EP 3020198**

54 Título: **Predicción de paletas en la codificación de vídeo basada en paleta**

30 Prioridad:

12.07.2013 US 201361845824 P
01.11.2013 US 201361899048 P
06.12.2013 US 201361913040 P
10.07.2014 US 201414328502

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
26.09.2017

73 Titular/es:

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

72 Inventor/es:

GUO, LIWEI;
KARCZEWICZ, MARTA;
SOLE ROJALS, JOEL;
JOSHI, RAJAN LAXMAN;
KIM, WOO-SHIK y
PU, WEI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 633 947 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

Predicción de paletas en la codificación de vídeo basada en paleta

5 CAMPO TÉCNICO

La presente divulgación se refiere a la codificación y decodificación de vídeo.

ANTECEDENTES

10 Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión digital directa, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo en continuo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de vídeo (AVC), la norma de Codificación de vídeo de Alta Eficiencia (HEVC) actualmente en desarrollo y las extensiones de dichas normas. 15 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. 20

Las técnicas de compresión de vídeo llevan a cabo la predicción espacial (intra-imagen) y/o la predicción temporal (entre imágenes) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo) puede repartirse en bloques de vídeo. Los bloques de vídeo en un fragmento intra-codificado (1) de una imagen se codifican usando la predicción espacial con respecto a muestras de referencia en bloques contiguos de la misma imagen. Los bloques de vídeo en un fragmento inter-codificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques contiguos de la misma imagen, o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia. 25 30

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque inter-codificado 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 que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intra-codificado se codifica de acuerdo con una modalidad de intra-codificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformación, dando como resultado coeficientes residuales, los cuales pueden cuantizarse posteriormente. Los coeficientes cuantizados, inicialmente dispuestos en una matriz bidimensional, pueden explorarse con el fin de producir un vector unidimensional de coeficientes, y puede aplicarse la codificación por entropía para lograr aún más compresión. 35 40

Un flujo de bits de codificación multivista puede generarse codificando vistas, por ejemplo, desde múltiples perspectivas. Se han desarrollado algunas normas de vídeo tridimensional (3D) que hacen uso de aspectos de la codificación multivista. Por ejemplo, diferentes vistas pueden transmitir las vistas de los ojos izquierdo y derecho para admitir el vídeo 3D. De forma alternativa, algunos procesos de codificación de vídeo 3D pueden aplicar la denominada codificación multivista más profundidad. En la codificación multivista más profundidad, un flujo de bits de vídeo 3D puede contener no solo componentes de vistas de textura, sino también componentes de vistas de profundidad. Por ejemplo, cada vista puede comprender un componente de vista de textura y un componente de vista de profundidad. El documento US 7,343,037 B1 divulga un esquema de paletización dinámico, localmente adaptativo. Se codifica una lista de colores usados utilizando, si es posible, un índice en la memoria intermedia de colores de píxeles LRU (último usado recientemente) en lugar de codificar el color real. 45 50

El documento US 2011/0110416 A1 divulga otro sistema para codificar y decodificar vídeo e imágenes fijas en el que se emplea la codificación de paleta. Las paletas pueden codificarse diferencialmente, que es un formato de un bit por entrada de paleta más datos de paleta. Si el bit es uno, entonces la entrada es bruta, de lo contrario los datos son un índice en la tabla de la paleta anterior. 55

60 SUMARIO

Las técnicas de la presente divulgación se refieren a la codificación de vídeo basada en paleta. En la codificación basada en paleta, un codificador de vídeo (por ejemplo, un codificador de vídeo o un decodificador de vídeo) puede formar una denominada "paleta" como una tabla de colores o valores de píxel que representan los datos de vídeo de un área particular (por ejemplo, un bloque dado). De esta manera, en lugar de codificar valores de píxel reales o sus residuos para un bloque actual de datos de vídeo, el codificador de vídeo puede codificar valores de índice para 65

uno o más de los valores de píxel del bloque actual, donde los valores de índice indican entradas en la paleta que se usan para representar los valores de píxel del bloque actual. Una paleta actual para un bloque actual de datos de vídeo puede codificarse explícitamente y enviarse al descodificador de vídeo, predecirse a partir de entradas de paleta anteriores, predecirse a partir de valores de píxel anteriores, o una combinación de los mismos.

De acuerdo con las técnicas descritas en la presente divulgación para generar una paleta actual para un bloque actual, el descodificador de vídeo determina en primer lugar una o más entradas de paleta en una paleta predictiva que se copian a la paleta actual y, a continuación, determina un cierto número de entradas de paleta nuevas que no están en la paleta predictiva pero que se incluyen en la paleta actual. Basándose en esta información, el descodificador de vídeo calcula un tamaño de la paleta actual igual a la suma del número de las entradas de paleta copiadas y el número de las entradas de paleta nuevas, y genera la paleta actual del tamaño determinado incluyendo las entradas de paleta copiadas y las entradas de paleta nuevas. Un codificador de vídeo puede realizar técnicas similares para generar la paleta actual para el bloque actual. Además, el codificador de vídeo puede codificar y enviar explícitamente valores de píxel para las entradas de paleta nuevas al descodificador de vídeo. Las técnicas descritas en la presente divulgación también pueden incluir técnicas para varias combinaciones de uno o más modos de codificación basados en paletas de señalización, paletas de transmisión, paletas de predicción, paletas de deducción o mapas de codificación basados en paletas de transmisión y otros elementos sintácticos.

En un ejemplo, la presente divulgación está dirigida a un procedimiento de codificación de datos de vídeo como se define mediante las características de la reivindicación 1.

En otro ejemplo, la presente divulgación está dirigida a un aparato para codificar datos de vídeo de acuerdo con la reivindicación 13.

En un ejemplo adicional, la presente divulgación está dirigida a un medio no transitorio legible por ordenador que almacena instrucciones en el mismo de acuerdo con la reivindicación 14.

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

BREVE DESCRIPCIÓN DE LOS DIBUJOS

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

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

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

La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de determinación de una paleta para codificar datos de vídeo, coherente con las técnicas de la presente divulgación.

La FIG. 5 es un diagrama conceptual que ilustra ejemplos de determinación de índices para una paleta para un bloque de vídeo, coherente con las técnicas de la presente divulgación.

La FIG. 6 es un diagrama conceptual que ilustra ejemplos de determinación de un borde geométrico de un bloque de vídeo usando una serie de índices de paleta para el componente de luma, submuestreados de forma adaptativa para los componentes de croma, coherente con las técnicas de la presente divulgación.

La FIG. 7 es un diagrama de flujo que ilustra un proceso de ejemplo para codificar datos de vídeo residuales de predicción usando un modo de codificación basado en paleta, coherente con las técnicas de la presente divulgación.

La FIG. 8 es un diagrama de flujo que ilustra un proceso de ejemplo para descodificar datos de vídeo residuales de predicción usando un modo de codificación basado en paleta, coherente con las técnicas de la presente divulgación.

La FIG. 9 es un diagrama de flujo que ilustra un proceso de ejemplo para generar una paleta para la codificación basada en paleta, coherente con las técnicas de la presente divulgación.

DESCRIPCIÓN DETALLADA

La presente divulgación incluye técnicas para la codificación y compresión de vídeo. En particular, la presente

divulgación describe técnicas para codificación de datos de vídeo basada en paleta. En la codificación de vídeo tradicional, se supone que las imágenes son de tono continuo y espacialmente uniformes. Basándose en estas suposiciones, se han desarrollado varias herramientas tales como la transformación basada en bloques, el filtrado, etc., y estas herramientas han mostrado un buen rendimiento para los vídeos de contenido natural.

5 Sin embargo, en aplicaciones como escritorio remoto, trabajo en equipo y pantalla inalámbrica, el contenido de la 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 discontinuos y presentar líneas bien definidas, y límites de objetos de alto contraste. La suposición de tono continuo y uniformidad ya no puede aplicarse al contenido
10 de la pantalla, por lo que las técnicas tradicionales de codificación de vídeo pueden no ser formas eficientes de comprimir datos de vídeo que incluyen contenido de la pantalla.

La presente divulgación describe la codificación basada en paleta, que puede ser particularmente adecuada para la codificación de contenido generado en pantalla. Por ejemplo, suponiendo que un área particular de datos de vídeo
15 tiene un número relativamente pequeño de colores, un codificador de vídeo (un codificador de vídeo o descodificador de vídeo) puede formar una denominada "paleta" como una tabla de colores o valores de píxel que representan los datos de vídeo del área particular (por ejemplo, un bloque dado). Por ejemplo, la paleta puede incluir los valores de píxel más dominantes en el bloque dado. En algunos casos, los valores de píxel más dominantes pueden incluir el uno o más valores de píxel que ocurren con más frecuencia dentro del bloque. Además, en algunos
20 casos puede aplicarse un valor umbral para definir si se incluye un valor de píxel como uno de los valores de píxel más dominantes en el bloque. De acuerdo con la presente divulgación, en lugar de codificar valores de píxel reales o sus residuos para un bloque actual de datos de vídeo, el codificador de vídeo puede codificar valores de índice indicativos de uno o más de los valores de píxel del bloque actual, donde los valores de índice indican entradas en la paleta que se usan para representar los valores de píxel del bloque actual.

25 Por ejemplo, el codificador de vídeo puede codificar un bloque de datos de vídeo determinando la paleta para el bloque (por ejemplo, codificando la paleta explícitamente, prediciendo la paleta, o una combinación de los mismos), localizando una entrada en la paleta para representar uno o más de los valores de píxel, y codificando el bloque con valores de índice que indican la entrada en la paleta usada para representar los valores de píxel del bloque. En
30 algunos ejemplos, el codificador de vídeo puede señalar los valores de índice en un flujo de bits codificado. Un descodificador de vídeo puede obtener, a partir de un flujo de bits codificado, una paleta para un bloque, así como valores de índice para los píxeles del bloque. El descodificador de vídeo puede relacionar los valores de índice de los píxeles con las entradas de paleta para reconstruir los valores de píxel del bloque.

35 Los ejemplos anteriores pretenden proporcionar una descripción general de la codificación basada en paleta. En varios ejemplos, las técnicas descritas en la presente divulgación pueden incluir técnicas para varias combinaciones de uno o más modos de codificación basados en paletas de señalización, paletas de transmisión, paletas de predicción, paletas de deducción o mapas de codificación basados en paletas de transmisión, y otros elementos sintácticos. Dichas técnicas pueden mejorar la eficiencia de la codificación de vídeo, por ejemplo, requiriendo menos
40 bits para representar el contenido generado en pantalla.

Por ejemplo, una paleta actual para un bloque actual de datos de vídeo puede codificarse explícitamente y enviarse al descodificador de vídeo, predecirse a partir de entradas de paleta anteriores, predecirse a partir de valores de píxel anteriores, o una combinación de los mismos. De acuerdo con las técnicas descritas en la presente divulgación
45 para generar una paleta actual para un bloque actual, el descodificador de vídeo determina en primer lugar una o más entradas de paleta en una paleta predictiva que se copian a la paleta actual y, a continuación, determina un cierto número de entradas de paleta nuevas que no están en la paleta predictiva pero que se incluyen en la paleta actual. Basándose en esta información, el descodificador de vídeo calcula un tamaño de la paleta actual igual a la suma del número de las entradas de paleta copiadas y el número de las entradas de paleta nuevas, y genera la
50 paleta actual del tamaño determinado incluyendo las entradas de paleta copiadas y las entradas de paleta nuevas. Un codificador de vídeo puede realizar técnicas similares para generar la paleta actual para el bloque actual. Además, el codificador de vídeo puede codificar y enviar explícitamente valores de píxel para las entradas de paleta nuevas al descodificador de vídeo.

55 En algunos ejemplos de la presente divulgación, las técnicas para la codificación basada en paleta de datos de vídeo pueden usarse con una o más técnicas de codificación diferentes, tales como técnicas para la codificación inter-predictiva o la codificación intra-predictiva de datos de vídeo. Por ejemplo, como se describe con mayor detalle a continuación, se puede configurar un codificador o descodificador, o un codificador-descodificador combinado (codec), para realizar codificación inter e intra-predictiva, así como codificación basada en paleta. En algunos
60 ejemplos, las técnicas de codificación basadas en paleta pueden configurarse para su uso en uno o más modos de unidades de codificación (CU) de Codificación de vídeo de alta eficiencia (HEVC). En otros ejemplos, las técnicas de codificación basadas en paleta pueden usarse de forma independiente o como parte de otros sistemas o normas existentes o futuros.

65 La Codificación de vídeo de alta eficiencia (HEVC) es una nueva norma de codificación de vídeo desarrollada por el Equipo de colaboración conjunta en 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. Un borrador reciente de la norma HEVC, denominado "Borrador 10 de HEVC" o "WD10", se describe en el documento JCTVC-L1003v34, Bross et al., "High Efficiency Video Coding (HEVC) Text Specification Draft 10 (for FDIS & Last Call)"["Memoria descriptiva textual de la Codificación de vídeo de alta eficiencia (HEVC), Borrador 10 (para FDIS (Borrador final de la norma internacional) y Último aviso)], Equipo de colaboración conjunta en codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12ª conferencia: Ginebra, Suiza, 14 a 23 de enero de 2013, disponible en http://phenix.intevry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip.

Con respecto a la estructura de HEVC, como un ejemplo, las técnicas de codificación basadas en paleta pueden configurarse para usarse como un modo de CU. En otros ejemplos, las técnicas de codificación basadas en paleta pueden configurarse para usarse como un modo de PU en la estructura de HEVC. En consecuencia, todos los procesos divulgados siguientes descritos en el contexto de un modo de CU pueden, de forma adicional o de forma alternativa, aplicarse a PU. Sin embargo, estos ejemplos basados en HEVC no deben considerarse una restricción o limitación de las técnicas de codificación basadas en paleta descritas en el presente documento, ya que dichas técnicas pueden aplicarse para trabajar de forma independiente o como parte de otros sistemas/normas existentes o aún no desarrollados. En estos casos, la unidad para la codificación de paleta puede ser bloques cuadrados, bloques rectangulares o incluso regiones de forma no rectangular.

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo de ejemplo 10 que puede utilizar las técnicas de la presente divulgación. Tal como se utiliza en el presente documento, el término "codificador de vídeo" se refiere de forma genérica tanto a codificadores de vídeo como a descodificadores de vídeo. En la presente divulgación, los términos "codificación de vídeo" o "codificación" pueden referirse de forma genérica a codificación de vídeo o descodificación de vídeo. El codificador de vídeo 20 y el descodificador de vídeo 30 del sistema de codificación de vídeo 10 representan ejemplos de dispositivos que pueden configurarse para realizar técnicas para la codificación de vídeo basada en paleta de acuerdo con diversos ejemplos descritos en la presente divulgación. Por ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para codificar de forma selectiva varios bloques de datos de vídeo, tales como CU o PU en codificación HEVC, usando codificación basada en paleta o codificación no basada en paleta. Los modos de codificación no basados en paleta pueden referirse a varios modos de codificación temporal inter-predictiva o modos de codificación espacial intra-predictiva, tales como los diversos modos de codificación especificados por el Borrador 10 de HEVC.

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 un dispositivo de codificación de vídeo o un 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 puede denominarse un dispositivo de descodificación de vídeo o un 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.

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

El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 a través de un canal 16. El enlace 16 puede comprender uno o más medios o dispositivos capaces de transferir los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicación que permiten al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de origen 12 puede modular los datos de vídeo codificados de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. El uno o más medios de comunicación pueden incluir medios de comunicación inalámbricos y/o cableados, tales como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. 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 extensa o una red global (por ejemplo, Internet). El uno o más medios de comunicación pueden incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que facilite la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

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 a través del acceso a disco o el acceso a tarjeta. El medio de almacenamiento puede incluir diversos medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, discos DVD, discos CD-ROM, memoria flash, u otros medios de almacenamiento digital adecuados para almacenar datos de vídeo codificados.

En un ejemplo adicional, 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 los datos de vídeo codificados almacenados en el servidor de archivos u otro dispositivo de almacenamiento intermedio a través de transmisión en continuo o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir los datos de vídeo codificados al dispositivo de destino 14. Los servidores de archivos de ejemplo incluyen servidores web (por ejemplo, para un sitio web), servidores de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados a la red (NAS) y unidades de disco local.

El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de una conexión de datos estándar, tal como una conexión a Internet. Entre los ejemplos de tipos de conexiones de datos pueden incluirse canales inalámbricos (por ejemplo, conexiones Wi-Fi), conexiones por cable (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambas que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión en continuo, una transmisión de descarga o una combinación de ambas.

Las técnicas de la presente divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo como apoyo a diversas aplicaciones multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, mediante Internet, codificación de datos de vídeo para su almacenamiento en un medio de almacenamiento de datos, descodificación de datos de vídeo almacenados en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 puede configurarse para admitir transmisión de vídeo unidireccional o bidireccional, para admitir aplicaciones tales como la transmisión de vídeo en continuo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

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

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye un origen 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. El origen de vídeo 18 puede incluir un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene datos de vídeo capturados previamente, una interfaz de alimentación de vídeo para recibir datos de vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos de ordenador para generar datos de vídeo, o una combinación de dichos orígenes de datos de vídeo.

El codificador de vídeo 20 puede codificar datos de vídeo del origen de vídeo 18. En algunos ejemplos, el dispositivo de origen 12 transmite directamente los datos de vídeo codificados al dispositivo destino 14 a través de la interfaz de salida 22. En otros ejemplos, los datos de vídeo codificados también pueden almacenarse en un medio de almacenamiento o en un servidor de archivos para que el dispositivo de destino 14 acceda posteriormente para la descodificación y/o reproducción.

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 sobre 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 datos de vídeo descodificados. El dispositivo de visualización 32 puede comprender diversos 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.

La presente 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 "señalización" o "transmisión" 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 ocurrir en tiempo real o casi real. De forma alternativa, dicha comunicación puede ocurrir durante un intervalo de tiempo, tal como podría ocurrir cuando se almacenan elementos sintácticos en un medio de almacenamiento legible por ordenador en un flujo de bits codificados en el instante de la codificación, que pueden recuperarse mediante un dispositivo de descodificación en cualquier instante después de almacenarse en este medio. Así pues, aunque el descodificador de vídeo 30 puede denominarse "receptor" de cierta información, la recepción de información no ocurre necesariamente en tiempo real o casi real y puede recuperarse de un medio en algún instante después del almacenamiento.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse como cualquiera de diversos circuitos adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables por campo (FPGA), lógica discreta, hardware, o cualquier combinación de estos. Si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en unos medios de almacenamiento legibles por ordenador no transitorios adecuados, y puede ejecutar las instrucciones en hardware usando uno o más procesadores que realizan las técnicas de la presente divulgación. Cualquiera de los anteriores (incluyendo hardware, software, una combinación de hardware y software, etc.) puede considerarse como uno o más procesadores. Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o decodificadores, donde cualquiera de ambos puede estar integrado como parte de un codificador-decodificador combinado (CDEC) en un dispositivo respectivo.

En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 funcionan de acuerdo con una norma de compresión de vídeo, tal como la norma HEVC mencionada anteriormente, y descrita en el Borrador 10 de HEVC. Además de la norma HEVC básica, hay iniciativas en progreso para producir codificación de vídeo escalable, codificación de vídeo multivista y ampliaciones de la codificación 3D para HEVC. Además, se pueden proporcionar modos de codificación basados en paleta, por ejemplo, como se describe en la presente divulgación, para la ampliación de la norma HEVC. En algunos ejemplos, las técnicas descritas en la presente divulgación para la codificación basada en paleta se pueden aplicar a codificadores y decodificadores configurados para funcionar de acuerdo con otras normas de codificación de vídeo, tales como la norma ITU-T-H.264/AVC o normas futuras. En consecuencia, se describe a modo de ejemplo la aplicación de un modo de codificación basado en paleta para la codificación de unidades de codificación (CU) o unidades de predicción (PU) en un codec HEVC.

En HEVC y otras normas de codificación de vídeo, una secuencia de vídeo incluye típicamente una serie de imágenes. Las imágenes también pueden denominarse "tramas". Una imagen puede incluir tres matrices de muestras, denominadas 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 pueden denominarse en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede incluir únicamente una matriz de muestras de luma.

Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades de árbol de codificación (CTU). Cada una de las CTU puede ser un bloque de árbol de codificación de muestras de luma, dos bloques de árbol de codificación correspondientes de muestras de croma, y estructuras sintácticas usadas para codificar las muestras de los bloques de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque de muestras $N \times N$. Una CTU también puede denominarse un "bloque de árbol" o una "unidad de codificación más grande" (LCU). Las CTU de HEVC pueden ser análogas en términos generales a los macrobloques de otras normas, tales como H.264/AVC. Sin embargo, una CTU no está limitada necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un fragmento puede incluir un número entero de CTU ordenadas consecutivamente en la exploración de trama. Un fragmento codificado puede comprender una cabecera de fragmento y datos de fragmento. La cabecera de fragmento de un fragmento puede ser una estructura sintáctica que incluye elementos sintácticos que proporcionan información sobre el fragmento. Los datos del fragmento pueden incluir CTU codificadas del fragmento.

La presente divulgación puede usar el término "unidad de vídeo" o "bloque de vídeo" o "bloque" para referirse a uno o más bloques de muestras y estructuras sintácticas usadas para codificar muestras de uno o más bloques de muestras. Los tipos de ejemplos de unidades o bloques de vídeo pueden incluir CTU, CU, PU, unidades de transformación (TU), macrobloques, reparticiones de macrobloques, etc. En algunos contextos, el análisis de las PU puede intercambiarse con el análisis de macrobloques o reparticiones de macrobloques.

Para generar una CTU codificada, el codificador de vídeo 20 puede realizar de forma recursiva la repartición de árbol cuádruple 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 $N \times N$. Una CU puede ser un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene una matriz de muestras de luma, una matriz de muestras de Cb y una matriz de muestras de Cr, y estructuras sintácticas usadas para codificar las muestras de los bloques de codificación. El codificador de vídeo 20 puede repartir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción puede ser un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las que se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede ser un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma de una imagen, y estructuras sintácticas usadas para predecir las muestras de bloques de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU.

El codificador de vídeo 20 puede usar intra-predicción o inter-predicción para generar los bloques predictivos para

una PU. Si el codificador de vídeo 20 usa la intra-predicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU basándose en muestras descodificadas de la imagen asociada con la PU.

- 5 Si el codificador de vídeo 20 usa la inter-predicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU basándose en muestras descodificadas de una o más imágenes distintas de la imagen asociada con la PU. El codificador de vídeo 20 puede usar uni-predicción o bi-predicción para generar los bloques predictivos de una PU. Cuando el codificador de vídeo 20 usa uni-predicción 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 bi-predicción para generar los bloques predictivos para una PU, la PU puede tener dos MV.

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

Asimismo, el codificador de vídeo 20 puede usar repartición de árbol cuádruple para descomponer los bloques residuales (por ejemplo, bloques residuales de luma, Cb y Cr) de una CU en uno o más bloques de transformación (por ejemplo, bloques de transformación de luma, Cb y Cr). Un bloque de transformación puede ser un bloque rectangular de muestras en el que se aplica la misma transformación. Una unidad de transformación (TU) de una CU puede ser un bloque de transformación de muestras de luma, dos bloques de transformación correspondientes de muestras de croma, y estructuras sintácticas usadas para transformar las muestras de bloques de transformación. Así pues, cada TU de una CU puede asociarse con un bloque de transformación de luma, un bloque de transformación de Cb y un bloque de transformación de Cr. El bloque de transformación de luma asociado con la TU puede ser un sub-bloque del bloque residual luma de la CU. El bloque de transformación de Cb puede ser un sub-bloque del bloque residual de Cb de la CU. El bloque de transformación de Cr puede ser un sub-bloque del bloque residual de Cr de la CU.

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

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 cuantizar el bloque de coeficientes. La cuantización se refiere en general a un proceso en el que los coeficientes de transformación se cuantizan para reducir posiblemente la cantidad de datos usados para representar los coeficientes de transformación, proporcionando una compresión adicional. Después de que el codificador de vídeo 20 cuantiza un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos sintácticos que indican los coeficientes de transformación cuantizados. Por ejemplo, el codificador de vídeo 20 puede realizar una Codificación aritmética binaria adaptable al contexto (CABAC) en los elementos sintácticos que indican los coeficientes de transformación cuantizados. El codificador de vídeo 20 puede enviar los elementos sintácticos codificados por entropía en un flujo de bits. El flujo de bits también puede incluir elementos sintácticos que no están codificados por entropía.

El codificador de vídeo 20 puede enviar un flujo de bits que incluye los elementos sintácticos codificados por entropía. El flujo de bits puede incluir una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL incluye una cabecera de la unidad NAL y encapsula una carga útil de secuencia de bytes en bruto (RBSP). La cabecera de la 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 la 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 se encapsula en una unidad NAL. En algunos casos, una RBSP incluye cero bits.

Diferentes tipos de unidades NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad NAL puede encapsular una RBSP para un conjunto de parámetros de imagen (PPS), un segundo tipo de unidad NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad NAL puede encapsular una RBSP para información de mejora suplementaria (SEI), y así sucesivamente. Las unidades NAL que encapsulan RBSP para datos de codificación de vídeo (a diferencia de RBSP para conjuntos de parámetros y mensajes SEI) se pueden denominar unidades NAL de la capa de codificación de vídeo (VCL).

El descodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede obtener elementos sintácticos del flujo de bits. Por ejemplo, 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 obtenidos (por ejemplo, descodificados) 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 MV de PU para determinar bloques de muestras predictivos (es decir, bloques predictivos) para las PU de una CU actual. Además, el descodificador de vídeo 30 puede cuantizar de forma inversa los bloques de coeficientes de transformación asociados con las TU de la CU actual. El descodificador de vídeo 30 puede realizar transformaciones inversas en los bloques de coeficientes de transformación para reconstruir los bloques de transformación asociados con 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 de muestras predictivos para las PU de la CU actual a muestras correspondientes de los bloques de transformación de las TU de la CU actual. Mediante la reconstrucción de los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar una codificación basada en paleta. Por ejemplo, en la codificación basada en paleta, en lugar de realizar las técnicas de codificación intra-predictiva o inter-predictiva descritas anteriormente, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar una denominada paleta como una tabla de colores o valores de píxel que representan los datos de vídeo de un área particular (por ejemplo, un bloque dado). De esta manera, en lugar de codificar valores de píxel reales o sus residuos para un bloque actual de datos de vídeo, el codificador de vídeo puede codificar valores de índice para uno o más de los valores de píxel del bloque actual, donde los valores de índice indican entradas en la paleta que se usan para representar los valores de píxel del bloque actual.

En un ejemplo, el codificador de vídeo 20 puede codificar un bloque de datos de vídeo determinando una paleta para el bloque, localizando una entrada en la paleta que tiene un valor representativo del valor de uno o más píxeles del bloque, y codificando el bloque con valores de índice que indican la entrada en la paleta usada para representar el uno o más valores de píxel del bloque. En algunos ejemplos, el codificador de vídeo 20 puede señalar los valores de índice en un flujo de bits codificado. Un descodificador de vídeo puede obtener, a partir de un flujo de bits codificado, una paleta para un bloque, así como valores de índice para los píxeles del bloque. El descodificador de vídeo puede relacionar los valores de índice de los píxeles con entradas de paleta para reconstruir los valores de píxel del bloque.

En otro ejemplo, el codificador de vídeo 20 puede codificar un bloque de datos de vídeo determinando valores residuales de predicción para el bloque, determinando una paleta para el bloque, localizando una entrada en la paleta que tiene un valor representativo del valor de uno o más de los valores residuales de predicción, y codificando el bloque con valores de índice que indican la entrada en la paleta usada para representar los valores residuales de predicción para el bloque. El descodificador de vídeo 30 puede obtener, a partir de un flujo de bits codificado, una paleta para un bloque, así como valores de índice para los valores residuales de predicción del bloque. El descodificador de vídeo 30 puede relacionar los valores de índice de los valores residuales de predicción con entradas de paleta para reconstruir los valores residuales de predicción del bloque. Los valores residuales de predicción pueden añadirse a los valores de predicción (por ejemplo, obtenidos usando intra o inter-predicción) para reconstruir los valores de píxel del bloque.

Como se describe con más detalle a continuación, la idea básica de la codificación basada en paleta es que, para un bloque dado de datos de vídeo a codificar, se obtiene una paleta que incluye los valores de píxel más dominantes en el bloque actual. Por ejemplo, la paleta puede referirse a una serie de valores de píxel que se supone que son dominantes y/o representativos de la CU actual. El codificador de vídeo 20 puede transmitir en primer lugar el tamaño y los elementos de la paleta al descodificador de vídeo 30. El codificador de vídeo 20 puede codificar los valores de píxel en el bloque dado de acuerdo con un cierto orden de exploración. Para cada localización de píxel en el bloque dado, el codificador de vídeo 20 puede transmitir un indicador u otro elemento sintáctico para indicar si el valor de píxel en la localización de píxel está incluido en la paleta o no. Si el valor de píxel está en la paleta (es decir, existe una entrada de paleta que especifica el valor de píxel), el codificador de vídeo 20 puede señalar el valor de índice asociado con el valor de píxel para la localización de píxel en el bloque dado seguido de una "serie" de valores de píxel consecutivos del mismo valor en el bloque dado. En este caso, el codificador de vídeo 20 no transmite el indicador o el índice de paleta para las siguientes localizaciones de píxel que están incluidas en la

"serie", ya que todas ellas tienen el mismo valor de píxel.

Si el valor de píxel no está en la paleta (es decir, no existe ninguna entrada de paleta que especifique el valor de píxel), el codificador de vídeo 20 puede transmitir el valor de píxel o un valor residual (o sus versiones cuantizadas) para la localización de píxel dada en el bloque dado. El codificador de vídeo 30 puede determinar en primer lugar la paleta basándose en la información recibida del codificador de vídeo 20. El descodificador de vídeo 30 puede entonces asignar los valores de índice recibidos asociados con las localizaciones de píxel en el bloque dado a entradas de paleta para reconstruir los valores de píxel del bloque dado.

La codificación basada en paleta puede tener una cierta cantidad de sobrecarga de señalización. Por ejemplo, puede ser necesario un cierto número de bits para señalar las características de una paleta, tal como un tamaño de la paleta, así como la propia paleta. Además, puede ser necesario un cierto número de bits para señalar valores de índice para los píxeles del bloque. Las técnicas de la presente divulgación pueden, en algunos ejemplos, reducir el número de bits necesarios para señalar dicha información. Por ejemplo, las técnicas descritas en la presente divulgación pueden incluir técnicas para varias combinaciones de uno o más modos de codificación basados en paletas de señalización, paletas de transmisión, paletas de predicción, paletas de deducción, o mapas de codificación basados en paletas de transmisión y otros elementos sintácticos. Pueden implementarse técnicas particulares de la presente divulgación en el codificador de vídeo 20 y/o el descodificador de vídeo 30.

Aspectos de la presente divulgación están dirigidos a la predicción de paletas. Por ejemplo, de acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar una primera paleta que tiene un primer conjunto de entradas indicativas de los primeros valores de píxel. El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden entonces determinar, basándose en el primer conjunto de entradas de la primera paleta, un segundo conjunto de entradas indicativas de los segundos valores de píxel de una segunda paleta. El codificador de vídeo 20 y/o el descodificador de vídeo 30 también pueden codificar píxeles de un bloque de datos de vídeo usando la segunda paleta (es decir, usando el segundo conjunto de valores de píxel).

Cuando se determina el segundo conjunto de entradas de la segunda paleta basándose en el primer conjunto de entradas, el codificador de vídeo 20 puede codificar diversos elementos sintácticos, que el descodificador de vídeo 30 puede usar para reconstruir la segunda paleta. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos en un flujo de bits para indicar que una paleta completa (o paletas, en el caso de cada componente de color, por ejemplo Y, Cb, Cr o Y, U, V o R, G, B, de los datos de vídeo que tienen una paleta independiente) se predice a partir de (por ejemplo, se copia de) uno o más bloques contiguos del bloque que se está codificando actualmente.

La paleta a partir de la cual se predicen (por ejemplo, se copian) las entradas de paleta actual del bloque actual puede denominarse paleta predictiva. La paleta predictiva puede contener entradas de paleta de uno o más bloques contiguos incluyendo bloques espacialmente contiguos y/o bloques contiguos en un orden particular de exploración de los bloques. Por ejemplo, los bloques contiguos pueden estar localizados espacialmente a la izquierda (bloque contiguo izquierdo) o encima (bloque contiguo superior) del bloque que se está codificando actualmente. En otro ejemplo, el codificador de vídeo 20 puede determinar entradas de paleta predictiva usando los valores de las muestras más frecuentes en un entorno causal del bloque actual. En otro ejemplo, los bloques contiguos pueden ser contiguos al bloque que se está codificando actualmente de acuerdo con un orden de exploración particular usado para codificar los bloques. Es decir, los bloques contiguos pueden ser uno o más bloques codificados antes del bloque actual en el orden de exploración. El codificador de vídeo 20 puede codificar uno o más elementos sintácticos para indicar la localización de los bloques contiguos a partir de los que se copian las paletas.

En algunos ejemplos, la predicción de paletas se puede realizar por entrada. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos para indicar, para cada entrada de una paleta predictiva, si la entrada de paleta dada se incluye en la paleta actual para el bloque actual. Si el codificador de vídeo 20 no utiliza la predicción para propagar una entrada de la paleta actual para el bloque actual, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos adicionales para especificar las entradas no predichas, así como el número de dichas entradas, en la paleta actual para el bloque actual.

Como se ha descrito anteriormente, para un bloque actual, por ejemplo, una CU o PU, las entradas en su paleta pueden predecirse a partir de entradas en una paleta predictiva que incluyen entradas de paleta de uno o más bloques contiguos codificados previamente. La presente divulgación describe varias técnicas alternativas para estimar la paleta para el bloque actual.

En un ejemplo, una paleta predictiva incluye un cierto número de entradas, N. En este ejemplo, el codificador de vídeo 20 transmite en primer lugar un vector binario, V, que tiene el mismo tamaño que la paleta predictiva, es decir, un vector de tamaño N, al descodificador de vídeo 30. Cada entrada en el vector binario indica si la entrada correspondiente en la paleta predictiva se reutilizará o se copiará en una paleta actual para un bloque actual. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos que incluyen el vector binario. En algunos casos, el codificador de vídeo 20 codifica el vector binario que incluye un indicador de un bit para cada

una de las entradas de paleta en la paleta predictiva que indica si una entrada de paleta respectiva se copia en la paleta actual. En otros casos, el codificador de vídeo 20 codifica un vector binario comprimido sin pérdidas en el que las indicaciones para las entradas en el vector binario se comprimen o se combinan en conjunto en lugar de enviarse de forma individual como indicadores de un bit. De esta manera, el descodificador de vídeo 30 determina la una o más de las entradas de paleta en la paleta predictiva que se copian en la paleta actual.

Además, el codificador de vídeo 20 transmite un número, M, que indica cuántas entradas nuevas se incluirán en la paleta para el bloque actual y, a continuación, transmite valores de píxel para las nuevas entradas al descodificador de vídeo 30. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos que indican el número de las entradas de paleta nuevas que se incluyen en la paleta actual usando uno de códigos unarios, códigos unarios truncados, códigos de Golomb exponenciales, o códigos de Golomb-Rice. De esta manera, el descodificador de vídeo 30 determina el número de entradas de paleta nuevas que no están en la paleta predictiva que se incluyen en la paleta actual para el bloque actual.

En este ejemplo, el tamaño final de la paleta actual para el bloque actual puede obtenerse como igual a $M + S$, donde S es el número de entradas en la paleta predictiva que se reutilizan en la paleta para el bloque actual. El descodificador de vídeo 30 puede calcular un tamaño de la paleta actual igual a la suma de un número de las entradas de paleta copiadas y el número de las entradas de paleta nuevas. Una vez que se determina el tamaño de la paleta actual, el descodificador de vídeo 30 genera la paleta actual que incluye las entradas de paleta copiadas de la paleta predictiva y las entradas de paleta nuevas señalizadas explícitamente desde el codificador de vídeo 20.

Para generar la paleta para el bloque actual, el descodificador de vídeo 30 puede combinar las M entradas de paleta nuevas recibidas y las S entradas de paleta copiadas que se están reutilizando de la paleta predictiva. En algunos casos, la combinación puede basarse en los valores de píxel, de tal manera que las entradas en la paleta para el bloque actual pueden aumentar (o disminuir) con el índice de paleta, por ejemplo, cuando se usa una paleta independiente para cada componente. En otros casos, la combinación puede ser una concatenación de los dos conjuntos de entradas, es decir, las entradas de paleta copiadas y las entradas de paleta nuevas.

En un ejemplo que no forma parte de la invención, el codificador de vídeo 20 transmite en primer lugar una indicación de un tamaño de una paleta, N, para un bloque actual al descodificador de vídeo 30. El codificador de vídeo 20 transmite entonces un vector, V, que tiene el mismo tamaño que la paleta para el bloque actual, es decir, un vector de tamaño N, al descodificador de vídeo 30. Cada entrada en el vector indica si la entrada correspondiente en la paleta para el bloque actual se transmite explícitamente mediante el codificador de vídeo 20 o se copia de una paleta predictiva. Para las entradas que se copian de la paleta predictiva, el codificador de vídeo 20 puede usar diferentes procedimientos para señalar qué entrada en la paleta predictiva se usa en la paleta para el bloque actual. En algunos casos, el codificador de vídeo 20 puede señalar el índice de paleta que indica la entrada a copiar de la paleta predictiva en la paleta para el bloque actual. En otros casos, el codificador de vídeo 20 puede señalar un desplazamiento del índice, que es la diferencia entre el índice en la paleta para el bloque actual y el índice en la paleta predictiva.

En los dos ejemplos anteriores, el uno o más bloques contiguos codificados previamente, a partir de los cuales se forma la paleta predictiva usada para la predicción de la paleta actual para el bloque actual, pueden ser bloques espacialmente contiguos del bloque actual y/o bloques contiguos del bloque actual en un orden de exploración particular de los bloques. Por ejemplo, los bloques contiguos pueden estar espacialmente localizados encima (es decir, bloques contiguos superiores) o a la izquierda (es decir, bloques contiguos izquierdos) del bloque actual. En algunos ejemplos, puede construirse una lista de candidatos de bloques contiguos, y el codificador de vídeo 20 transmite un índice para indicar que uno o más de los bloques contiguos candidatos y las paletas asociadas se usan para formar la paleta predictiva.

Para ciertos bloques, por ejemplo, las CU al comienzo de un fragmento o en otros límites de fragmento o las CU del extremo izquierdo del fragmento o una imagen de datos de vídeo, la predicción de paletas puede estar deshabilitada. Por ejemplo, cuando el bloque actual de datos de vídeo comprende uno o un primer bloque en un fragmento de datos de vídeo o un bloque del extremo izquierdo del fragmento o una imagen de los datos de vídeo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden deshabilitar la copia de entradas de paleta en la paleta de predicción a la paleta actual para el bloque actual.

En un ejemplo adicional, el codificador de vídeo 20 transmite una indicación de un número de entradas incluidas en una paleta para un bloque actual al descodificador de vídeo 30. A continuación, para cada una de las entradas de paleta, el codificador de vídeo 20 transmite un indicador u otro elemento sintáctico para indicar si la entrada de paleta de la paleta para el bloque actual se transmite explícitamente mediante el codificador de vídeo 20 o si la entrada de paleta se obtiene a partir de un píxel reconstruido previamente. Para cada una de las entradas de paleta de la paleta para el bloque actual que se obtienen a partir de un píxel reconstruido previamente, el codificador de vídeo 20 transmite otra indicación con respecto a una localización de píxel del píxel reconstruido en el bloque actual o una localización de píxel del píxel reconstruido en el bloque contiguo que corresponde a la entrada de paleta. En algunos casos, la indicación de la localización del píxel reconstruido puede ser un vector de desplazamiento con respecto a la posición superior izquierda del bloque actual. En otros casos, la indicación de la localización del píxel

reconstruido puede ser un índice en una lista de píxeles reconstruidos que se pueden usar para especificar la entrada de paleta para el bloque actual. Por ejemplo, esta lista puede incluir todos los píxeles de referencia que se pueden usar para la intra-predicción normal en HEVC.

5 En algunos ejemplos, las técnicas para predecir una paleta completa pueden combinarse con técnicas para predecir una o más entradas de una paleta. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos en un flujo de bits para indicar si la paleta actual está totalmente copiada de la paleta predictiva (por ejemplo, la paleta para el último bloque con codificación de paleta). Si este no es el caso, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos en un flujo de bits para indicar si se copia cada entrada en la paleta predictiva.

10 En algunos casos, el tamaño de la paleta puede ser un valor fijo especificado en la norma de codificación de vídeo aplicada por el codificador de vídeo 20 y el decodificador de vídeo 30, o puede señalizarse desde el codificador de vídeo 20 al decodificador de vídeo 30. En el caso en el que cada uno de los componentes de color tiene una paleta independiente, el codificador de vídeo 20 puede señalar por separado los tamaños para las diferentes paletas. En el caso de una única paleta para todos los componentes de color, el codificador de vídeo 20 puede codificar un único tamaño para la única paleta. En otro ejemplo, en lugar de señalar el número de entradas y los valores de paleta, el codificador de vídeo 20 puede señalar, después de señalar cada valor de paleta, un indicador para indicar si el valor de paleta señalado es la entrada de paleta final para la paleta. El codificador de vídeo 20 puede no señalar un indicador de "fin de paleta" si la paleta ya ha alcanzado un cierto tamaño máximo.

25 El codificador de vídeo 20 puede codificar uno o más elementos sintácticos para indicar si la predicción de paletas está habilitada y/o activa. En un ejemplo con fines ilustrativos, el codificador de vídeo 20 puede codificar un indicador_paleta_pred para indicar, para cada bloque (por ejemplo, CU o PU), si el codificador de vídeo 20 usa la predicción de paletas para predecir la paleta para el bloque respectivo. En algunos ejemplos, el codificador de vídeo puede señalar un indicador independiente para cada componente de color (por ejemplo, tres indicadores para cada bloque). En otros ejemplos, el codificador de vídeo 20 puede señalar un indicador único que es aplicable a todos los componentes de color de un bloque.

30 El decodificador de vídeo 30 puede obtener la información anteriormente identificada a partir de un flujo de bits codificado y puede usar la información para reconstruir la paleta. Por ejemplo, el decodificador de vídeo 30 puede recibir datos que indican si una paleta particular se predice a partir de otra paleta, así como información que permite al decodificador de vídeo 30 usar las entradas de paleta predictiva apropiadas.

35 En algunos casos, de forma adicional o de forma alternativa, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden construir una paleta "sobre la marcha", es decir, de forma dinámica. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden añadir entradas a una paleta vacía durante la codificación. 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 un bloque. Los píxeles (por ejemplo, píxeles que tienen valores que se han añadido e indexado previamente dentro de la paleta) que se codifican relativamente más tarde en el bloque pueden referirse a entradas añadidas anteriormente de la paleta, por ejemplo, con valores de índice asociados con valores de píxel, en lugar de transmitir los valores de píxel. Igualmente, tras recibir un nuevo valor de píxel para una posición en un bloque, el decodificador de vídeo 30 puede seguir el mismo proceso que el codificador de vídeo 20 e incluir el valor de píxel en una paleta. De esta manera, el decodificador de vídeo 30 construye la misma paleta que el codificador de vídeo 20. El decodificador de vídeo 30 puede recibir, para los píxeles que tienen valores que ya están incluidos en la paleta, valores de índice que identifican los valores de píxel. El decodificador de vídeo 30 puede usar la información recibida, por ejemplo, valores de píxel para los valores de paleta y/o de índice, para reconstruir los píxeles de un bloque.

50 En algunos casos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden mantener una paleta de un tamaño fijo. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden añadir los valores de píxel reconstruidos más recientes a la paleta. Para cada entrada que se añade a la paleta, se descarta la entrada más antigua que se añadió a la paleta. Esto también se denomina a veces Primero en entrar, primero en salir (FIFO). Este proceso de actualización de la paleta puede aplicarse únicamente a los bloques que se codifican usando el modo de paleta o a todos los bloques independientemente del modo de codificación.

60 Las técnicas descritas anteriormente se refieren en general al codificador de vídeo 20 y al decodificador de vídeo 30 que construyen y/o transmiten una paleta para la codificación basada en paleta. Otros aspectos de la presente divulgación se refieren a la construcción y/o transmisión de un mapa que permite que el codificador de vídeo 20 y/o el decodificador de vídeo 30 determinen valores de píxel. Por ejemplo, otros aspectos de la presente divulgación se refieren a la construcción y/o transmisión de un mapa de índices que indican entradas en una paleta que especifican valores de píxel de un bloque de datos de vídeo.

65 En algunos ejemplos, el codificador de vídeo 20 puede indicar si los píxeles de un bloque tienen un valor correspondiente en una paleta. En un ejemplo con fines ilustrativos, se supone que una entrada (i, j) de un mapa corresponde a una posición de píxel (i, j) en un bloque de datos de vídeo. En este ejemplo, el codificador de vídeo

20 puede codificar un indicador para cada posición de píxel de un bloque. El codificador de vídeo 20 puede establecer el indicador igual a uno para la entrada (i, j) para indicar que el valor de píxel en la localización (i, j) es uno de los valores de la paleta. Cuando se incluye un valor de píxel en la paleta (es decir, el indicador es igual a uno), el codificador de vídeo 20 también puede codificar datos que indican un índice de paleta para la entrada (i, j) que identifica la entrada correspondiente en la paleta que especifica el valor de píxel. Cuando un valor de píxel no se incluye en la paleta (es decir, el indicador es igual a cero), el codificador de vídeo 20 también puede codificar datos que indican un valor de muestra (posiblemente cuantizado) para el píxel. En algunos casos, el píxel que no está incluido en la paleta se denomina "píxel de escape".

El descodificador de vídeo 30 puede obtener los datos anteriormente descritos de un flujo de bits codificado y usar los datos para determinar un índice de paleta y/o un valor de píxel para una localización particular en un bloque. Por ejemplo, el descodificador de vídeo 30 puede descodificar uno o más elementos sintácticos que indican si cada uno de los valores de píxel del bloque actual tiene un valor de píxel correspondiente en la paleta actual, descodificar uno o más elementos sintácticos que indican los valores de índice para el uno o más valores de píxel del bloque actual que tienen valores de píxel correspondientes en la paleta actual, y descodificar uno o más elementos sintácticos que indican los valores de píxel para el uno o más valores de píxel del bloque actual que no tienen un valor de píxel correspondiente en la paleta actual.

En algunos casos, puede haber una correlación entre el índice de paleta al que se asigna un píxel en una posición dada y la probabilidad de que un píxel contiguo se asigne al mismo índice de paleta. Es decir, cuando un píxel se asigna a un índice de paleta particular, puede haber una probabilidad relativamente alta de que uno o más píxeles contiguos (en términos de localización espacial) se asignen al mismo índice de paleta.

De acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar y codificar uno o más índices de un bloque de datos de vídeo relativos a uno o más índices del mismo bloque de datos de vídeo. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden configurarse para determinar un primer valor de índice asociado con un primer píxel en un bloque de datos de vídeo, donde el primer valor de índice relaciona un valor del primer píxel con una entrada de una paleta. El codificador de vídeo 20 y/o el descodificador de vídeo 30 también pueden configurarse para determinar, basándose en el primer valor de índice, uno o más segundos valores de índice asociados con uno o más segundos píxeles en el bloque de datos de vídeo, y codificar el primer y el uno o más segundos píxeles del bloque de datos de vídeo. Así pues, en este ejemplo, los índices de un mapa pueden codificarse en relación con uno o más índices del mapa diferentes.

En algunos ejemplos, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos que indican un cierto número de píxeles consecutivos en un orden de exploración dado que están asignados al mismo valor de índice. La cadena de valores de índice de valor similar se puede denominar en el presente documento una "serie". En algunos ejemplos, un valor de píxel puede estar asociado con exactamente un valor de índice en una paleta. En consecuencia, en algunos casos, una serie de valores también puede referirse a una cadena de valores de píxel de valor similar. En otros ejemplos, como se describe con respecto a la siguiente codificación con pérdidas, puede asignarse más de un valor de píxel al mismo valor de índice en una paleta. En dichos ejemplos, una serie de valores se refiere a valores de índice de valor similar. En este escenario, en el lado del descodificador, series de valores de índice de valor similar pueden corresponder a series de valores de píxel que corresponden a los valores de índice.

En un ejemplo con fines ilustrativos, si dos índices consecutivos en un orden de exploración dado tienen valores diferentes, la serie es igual a cero. Si dos índices consecutivos en un orden de exploración dado tienen el mismo valor pero el tercer índice en el orden de exploración tiene un valor diferente, la serie es igual a uno. El descodificador de vídeo 30 puede obtener los elementos sintácticos que indican una serie a partir de un flujo de bits codificado y puede usar los datos indicados por los elementos sintácticos para determinar el número de localizaciones de píxel consecutivas que tienen el mismo valor de índice.

En algunos ejemplos, todas las localizaciones de píxel en el bloque actual que tienen valores de píxel que están en la paleta para el bloque actual se codifican con un índice de paleta seguido por una "serie" del valor de píxel en localizaciones de píxel consecutivas. En el caso de que solo haya una entrada en la paleta, la transmisión del índice de paleta o la "serie" se puede omitir para el bloque actual. En el caso en que el valor de píxel en una de las localizaciones de píxel en el bloque actual no tenga una coincidencia exacta con un valor de píxel en la paleta, el codificador de vídeo 20 puede seleccionar una de las entradas de paleta que tiene el valor de píxel más próximo y calcular un error de predicción o valor residual entre el valor de píxel original y el valor de píxel de predicción incluido en la paleta. El codificador de vídeo 20 puede cuantizar, codificar y transmitir el valor residual para la localización de píxel al descodificador de vídeo 30.

El descodificador de vídeo 30 puede entonces obtener un valor de píxel en la localización de píxel basándose en el índice de paleta recibido correspondiente. El valor de píxel obtenido y el valor residual (recibidos del codificador de vídeo 20) se usan entonces para predecir el valor de píxel en la localización de píxel en el bloque actual. En un ejemplo, el valor residual se codifica usando un procedimiento HEVC especificado por el Borrador 10 de HEVC, tal como aplicando un árbol cuádruple residual (RQT) para transformar el valor residual, cuantizar los coeficientes de transformación y codificar por entropía los coeficientes de transformación cuantizados. En algunos casos, los valores

residuales pueden cuantizarse directamente sin aplicar una transformación. Como un ejemplo, el descodificador de vídeo 30 puede descodificar uno o más elementos sintácticos que indican los valores de índice para uno o más valores de píxel del bloque actual, donde los valores de índice identifican valores de píxel correspondientes en la paleta actual como valores de píxel de predicción, y descodificar uno o más elementos sintácticos que indican valores residuales entre uno o más valores de píxel del bloque actual y los valores de píxel de predicción identificados en la paleta actual. En algunos casos, los ejemplos anteriores pueden denominarse codificación con pérdidas.

De forma adicional o de forma alternativa, de acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden realizar copiado de líneas para una o más entradas de un mapa. Las entradas también se pueden denominar "posiciones" debido a la relación entre entradas del mapa y posiciones de píxel de un bloque. El copiado de líneas puede depender, en algunos ejemplos, de la dirección de exploración. Por ejemplo, el codificador de vídeo 20 puede indicar que un valor de píxel o un valor de mapa de índices para una posición particular en un bloque es igual al valor de píxel o índice en una línea por encima (por ejemplo, anterior) de la posición particular (para una exploración orientada horizontalmente) o en la columna a la izquierda (por ejemplo, anterior) de la posición particular (para una exploración orientada verticalmente). El codificador de vídeo 20 también puede indicar, como una serie, el número de valores o índices de píxel en el orden de exploración que son iguales a los valores o índices de píxel en la línea por encima o en la columna a la izquierda de la posición particular. En este ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden copiar valores de píxel o índice desde la línea contigua especificada (o columna para exploración vertical) y para el número especificado de entradas para la línea (o columna para exploración vertical) del bloque que se está codificando actualmente.

En algunos casos, la línea (o columna para exploración vertical) a partir de la cual se copian los valores puede ser directamente adyacente a, por ejemplo, por encima o a la izquierda de, la línea (o columna para exploración vertical) de la posición que se está codificando actualmente. En otros ejemplos, un cierto número de líneas del bloque se pueden almacenar en memoria intermedia mediante el codificador de vídeo 20 y/o el descodificador de vídeo 30, de tal manera que cualquiera del número de líneas del mapa se pueden usar como valores predictivos para una línea del mapa que se está codificando actualmente. Se pueden aplicar técnicas similares a columnas anteriores para una exploración vertical. En un ejemplo con fines ilustrativos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden configurarse para almacenar las cuatro filas de índices o valores de píxel anteriores antes de codificar la fila de píxeles actual. En este ejemplo, la fila predictiva (la fila a partir de la cual se copian índices o valores de píxel) se puede indicar en un flujo de bits con un código unario truncado u otros códigos tales como códigos unarios. Con respecto a un código unario truncado, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar un valor máximo para el código unario truncado basándose en un cálculo de la fila máxima (por ejemplo, índice_fila-1) para exploraciones horizontales o un cálculo de la columna máxima (por ejemplo, índice_columa-1) para exploraciones verticales. Además, también se puede incluir en el flujo de bits una indicación del número de posiciones de la fila predictiva que se copian. En algunos casos, si la línea (o la columna en el caso de exploraciones verticales) a partir de la que se predice una posición actual pertenece a otro bloque (por ejemplo, CU o CTU) dicha predicción puede estar deshabilitada.

Como otro ejemplo, el codificador de vídeo 20 puede señalar una instrucción, tal como "copiar desde la mitad izquierda de la línea superior" o "copiar desde la mitad derecha de la línea superior", indicando la línea contigua y el número o parte de entradas de la línea contigua a copiar a la línea del mapa que se está codificando actualmente. Como un ejemplo adicional, el mapa de valores de índice puede reordenarse antes de la codificación. Por ejemplo, el mapa de valores de índice se puede girar 90, 180 o 270 grados, o girarse de arriba a abajo o de izquierda a derecha para mejorar la eficiencia de codificación. Así pues, puede usarse cualquier exploración para convertir la matriz bidimensional de valores de píxel o índice en una matriz unidimensional.

Las técnicas para codificar las denominadas series de entradas pueden usarse conjuntamente con las técnicas de copiado de líneas descritas anteriormente. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos (por ejemplo, un indicador) que indican si el valor de una entrada en un mapa se obtiene a partir de una paleta o el valor de una entrada en el mapa se obtiene a partir de una línea codificada previamente en el mapa. El codificador de vídeo 20 también puede codificar uno o más elementos sintácticos que indican un valor de índice de una paleta o la localización de la entrada en la línea (la fila o la columna). El codificador de vídeo 20 también puede codificar uno o más elementos sintácticos que indican un número de entradas consecutivas que comparten el mismo valor. El descodificador de vídeo 30 puede obtener dicha información a partir de un flujo de bits codificado y usar la información para reconstruir los valores de mapa y de píxel para un bloque.

Como se ha indicado anteriormente, los índices de un mapa se exploran en un orden particular. De acuerdo con aspectos de la presente divulgación, la dirección de exploración puede ser vertical, horizontal o diagonal (por ejemplo, 45 grados o 135 grados en diagonal en bloque). En algunos ejemplos, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos para cada bloque indicando una dirección de exploración para explorar los índices del bloque. De forma adicional o de forma alternativa, la dirección de exploración puede ser un valor constante o puede señalizarse o inferirse basándose en la denominada información lateral tal como, por ejemplo, el tamaño del bloque, el espacio de color y/o el componente de color. El codificador de vídeo 20 puede especificar exploraciones para cada componente de color de un bloque. De forma alternativa, una exploración especificada

puede aplicarse a todos los componentes de color de un bloque.

En algunos ejemplos, el codificador de vídeo 20 tal vez no transmita series de valores de índice de valor similar en un orden de exploración dado al descodificador de vídeo 30. En cambio, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden obtener implícitamente los valores de las series para determinar las entradas del mapa. En este caso, el codificador de vídeo 20 puede señalar al descodificador de vídeo 30 que ocurre una serie de un valor de índice dado, pero puede no señalar un valor de la serie. Por ejemplo, el valor de una serie puede ser un valor constante o puede obtenerse basándose en información lateral para el bloque actual de datos de vídeo que se está codificando, tal como, por ejemplo, el tamaño del bloque. En el caso en el que el valor de una serie depende del tamaño del bloque, la serie puede ser igual a la anchura del bloque actual, la altura del bloque actual, la mitad de la anchura (o la mitad de la altura) del bloque actual, una fracción de la anchura y/o la altura del bloque actual, o un múltiplo de la anchura y/o la altura del bloque actual. En algunos ejemplos, el codificador de vídeo 20 puede señalar el valor de una serie al descodificador de vídeo 30 usando sintaxis de alto nivel. En algunos ejemplos, la frase "sintaxis de alto nivel" se refiere a la sintaxis en conjuntos de parámetros, por ejemplo, conjuntos de parámetros de imagen (PPS), conjuntos de parámetros de secuencia (SPS) y conjuntos de parámetros de vídeo (VPS), y cabeceras de fragmento.

De forma adicional o de forma alternativa, puede que el codificador de vídeo 20 ni siquiera necesite transmitir el mapa al descodificador de vídeo 30. En cambio, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden obtener implícitamente una posición o localización de inicio de cada serie de valores de índice incluidos en el mapa. En un ejemplo, la norma de codificación de vídeo aplicada por el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede determinar que una serie solo puede comenzar en ciertas localizaciones. Por ejemplo, la serie solo puede comenzar al principio de cada fila, o al principio de cada N filas del bloque actual. La localización de inicio puede ser diferente para diferentes direcciones de exploración. Por ejemplo, si se utiliza la exploración vertical, la serie solo puede comenzar al principio de una columna o al principio de cada N columnas del bloque actual. En otro ejemplo, la ubicación de inicio puede obtenerse dependiendo de la información lateral del bloque actual. En el caso en el que la localización de inicio de una serie depende del tamaño del bloque, la localización de inicio puede ser el punto medio de cada fila y/o cada columna del bloque actual, o una fracción de cada fila y/o columna del bloque actual. En algunos ejemplos, el codificador de vídeo 20 puede señalar la posición de inicio al descodificador de vídeo 30 usando sintaxis de alto nivel.

En algunos ejemplos, la obtención de la posición de inicio implícita y la obtención de la serie implícita, descritas cada una de ellas anteriormente, pueden combinarse. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar que una serie de valores de índice de valor similar en el mapa es igual a una distancia entre dos posiciones de inicio contiguas. En el caso en el que la posición de inicio es el comienzo (es decir, la primera posición) de cada fila del bloque actual, entonces el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar que la longitud de la serie es igual a la longitud de una fila completa del bloque actual.

En algunos casos, descritos con más detalle a continuación, se genera y comparte una paleta para múltiples componentes de color en el bloque actual. Por ejemplo, para cada localización de píxel en el bloque actual, los valores de píxel en tres componentes de color (por ejemplo, luma Y y ambos componentes de croma U y V) pueden formar un vector (es decir, un vector de color). A continuación, se puede formar una paleta seleccionando un cierto número de vectores para representar el bloque actual. Puede ser posible tener una paleta de valores de píxel para el componente de luma y otra paleta de valores de píxel para los componentes de croma. El copiado de líneas descrito con más detalle anteriormente también puede funcionar con una única paleta. Con una paleta compartida, una entrada de paleta puede ser una tripleta de (Y, U, V) o (Y, Cb, Cr) o (R, G, B). En este caso, el índice de paleta para cada localización de píxel se señala como igual al índice de paleta de la fila anterior, si la exploración es horizontal, o de la columna de la izquierda, si la exploración es vertical y, a continuación, el número asociado de índices de paleta también se copia de la fila o columna anterior en función de la serie.

En el caso de una paleta compartida para dos o más componentes de color o bien de paletas independientes para cada uno de los componentes de color, la información geométrica puede compartirse entre los componentes de color. Normalmente existe una alta correlación entre las localizaciones de los bordes de bloques colocados en diferentes componentes de color porque los componentes de croma pueden haberse submuestreado a partir de los componentes de luma de una manera predefinida, tal como un muestreo 4:2:2 o 4:2:0.

Por ejemplo, en la codificación basada en paletas, puede usarse la codificación de series para indicar información de geometría para el bloque actual porque un borde del bloque actual romperá la serie. En el caso del formato croma 4:4:4, la serie puede generarse una vez y usarse para todos los componentes de color. La serie puede generarse basándose en uno de los componentes de color, o la serie puede generarse usando más de uno de los componentes de color. En el caso del formato de croma 4:2:2 o del formato de croma 4:2:0, la serie usada para el componente de luma puede ser submuestreada para su aplicación a los componentes de croma

Las técnicas de la presente divulgación también incluyen otros aspectos de la codificación basada en paleta. Por ejemplo, de acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar uno o más elementos sintácticos para cada bloque para indicar que el bloque se codifica

usando un modo de codificación de paleta. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar un indicador de modo de paleta (indicador_modos_PLT) para indicar si se va a usar un modo de codificación basado en paleta para codificar un bloque particular. En este ejemplo, el codificador de vídeo 20 puede codificar un indicador_modos_PLT que es igual a uno para especificar que el bloque que se está codificando actualmente ("bloque actual") se codifica usando un modo de paleta. Un valor del indicador_modos_PLT igual a cero especifica que el bloque actual no se codifica usando el modo de paleta. En este caso, el descodificador de vídeo 30 puede obtener el indicador_modos_PLT a partir del flujo de bits codificado y aplicar el modo de codificación basado en paleta para descodificar el bloque. En casos en los que hay más de un modo de codificación basado en paleta disponible (por ejemplo, hay más de una técnica basada en paleta disponible para codificación), uno o más elementos sintácticos pueden indicar uno de una pluralidad de modos de paleta diferentes para el bloque.

En algunos casos, el codificador de vídeo 20 puede codificar un indicador_modos_PLT que es igual a cero para especificar que el bloque actual no se codifica usando un modo de paleta. En dichos casos, el codificador de vídeo 20 puede codificar el bloque usando cualquiera de diversos modos de codificación inter-predictiva, intra-predictiva u otros. Cuando el indicador_modos_PLT es igual a cero, el codificador de vídeo 20 puede transmitir información adicional (por ejemplo, elementos sintácticos) para indicar el modo específico que se usa para codificar el bloque respectivo. En algunos ejemplos, como se describe a continuación, el modo puede ser un modo de codificación HEVC, por ejemplo, un modo inter-predictivo o un modo intra-predictivo regular en la norma HEVC. El uso del indicador_modos_PLT se describe a modo de ejemplo. En otros ejemplos, pueden usarse otros elementos sintácticos tales como códigos de varios bits para indicar si el modo de codificación basado en paleta se va a usar para uno o más bloques, o para indicar cuál de una pluralidad de modos se va a utilizar.

Cuando se usa un modo de codificación basado en paleta, se transmite una paleta mediante el codificador de vídeo 20, por ejemplo, usando una o más de las técnicas descritas en el presente documento, en el flujo de bits de datos de vídeo codificados para su uso mediante el descodificador de vídeo 30. Una paleta puede transmitirse para cada bloque o puede compartirse entre un cierto número de bloques. La paleta puede referirse a un número de valores de píxel que son dominantes y/o representativos para el bloque.

El tamaño de la paleta, por ejemplo, en términos del número de valores de píxel que se incluyen en la paleta, puede fijarse o puede señalizarse usando uno o más elementos sintácticos en un flujo de bits codificado. Como se describe con más detalle a continuación, un valor de píxel puede estar compuesto de un cierto número de muestras, por ejemplo, dependiendo del espacio de color usado para la codificación. Por ejemplo, un valor de píxel puede incluir muestras de luma y crominancia (por ejemplo, muestras de luma, crominancia U y crominancia V (YUV) o luma, crominancia Cb y crominancia Cr (YCbCr)). En otro ejemplo, un valor de píxel puede incluir muestras rojas, verdes y azules (RGB). Como se describe en el presente documento, el término valor de píxel puede referirse en general a una o más de las muestras que contribuyen a un píxel. Es decir, el término valor de píxel no se refiere necesariamente a todas las muestras que contribuyen a un píxel, y puede usarse para describir un único valor de muestra que contribuye a un píxel.

En algunos ejemplos, una paleta se puede transmitir por separado para cada componente de color de un bloque particular. Por ejemplo, en el espacio de color YUV, puede haber una paleta para el componente Y (que representa los valores Y), otra paleta para el componente U (que representa los valores U) y otra paleta para el componente V (que representa los valores V). En otro ejemplo, una paleta puede incluir todos los componentes de un bloque particular. En este ejemplo, la i-ésima entrada en la paleta puede incluir tres valores (por ejemplo, Y_i , U_i , V_i). De acuerdo con aspectos de la presente divulgación, uno o más elementos sintácticos pueden indicar por separado el tamaño de la paleta para cada componente (por ejemplo, Y, U, V o similares). En otros ejemplos, puede usarse un tamaño único para todos los componentes, de tal manera que uno o más elementos sintácticos indican el tamaño de todos los componentes.

El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden realizar la codificación basada en paleta de una manera con pérdidas o sin pérdidas. Es decir, en algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar datos de vídeo sin pérdidas para un bloque usando entradas de paleta que coinciden con los valores de píxel del bloque (o enviando los valores de píxel reales si el valor de píxel no está incluido en la paleta). En otros ejemplos, como se ha descrito con mayor detalle con respecto a la FIG. 5 siguiente, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar datos de vídeo para un bloque usando entradas de paleta que no coinciden exactamente con los valores de píxel del bloque (codificación con pérdidas). De forma similar, si el valor de píxel real no se incluye en la paleta, el valor de píxel real puede cuantizarse de una manera con pérdidas.

De acuerdo con las técnicas descritas en la presente divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden realizar la codificación basada en paleta de bloques de vídeo predichos. En un ejemplo, el codificador de vídeo 20 obtiene en primer lugar una paleta para un bloque actual basándose en los valores de píxel en el bloque actual y, a continuación, asigna los valores de píxel en el bloque actual a índices de paleta para la codificación. La asignación puede ser de uno a uno (es decir, para la codificación sin pérdidas) o de varios a uno (es decir, para la codificación con pérdidas). El codificador de vídeo 20 también asigna valores de píxel de referencia en un bloque codificado previamente que se usarán para predecir los valores de píxel en el bloque actual. Una vez que

los valores de píxel del bloque actual se han asignado a índices de paleta, el codificador de vídeo 20 puede codificar el bloque actual con índices de paleta usando procedimientos de codificación regular, por ejemplo, codificación intra regular en la norma HEVC.

5 En el ejemplo anterior, el bloque actual con índices de paleta se trata como si el bloque actual fuera un bloque original con valores de píxel. Del mismo modo, los índices de paleta de los píxeles de referencia se usan para realizar la intra-predicción regular en el bloque actual con índices de paleta. El codificador de vídeo 20 transmite el error de predicción o los valores residuales al descodificador de vídeo 30. Después de codificar el bloque actual, el codificador de vídeo 20 convierte los índices de los píxeles de referencia, los píxeles de predicción y los valores
10 residuales a los valores de píxel para la reconstrucción del bloque actual y la predicción normal de bloques futuros. El descodificador de vídeo 30 puede obtener los valores residuales codificados para el bloque actual a partir del flujo de bits, y descodificar el bloque actual usando el procedimiento de descodificación regular para obtener el bloque actual con índices de paleta. El descodificador de vídeo 30 puede entonces determinar los valores de píxel del bloque actual basándose en los valores de píxel en la paleta que están asociados con los índices de paleta.

15 En otro ejemplo, el codificador de vídeo 20 puede generar una paleta para un bloque actual donde la paleta incluye entradas que indican valores residuales de predicción para el bloque dado. Los valores residuales de predicción para el bloque dado se pueden generar usando cualquier modo de predicción, por ejemplo, inter-predicción o intra-predicción regular en la norma HEVC. Los valores residuales de predicción para el bloque dado pueden ser valores de píxel residuales o valores de coeficientes de transformación residuales. En cualquier caso, los valores residuales de predicción pueden cuantizarse. En este ejemplo, el codificador de vídeo 20 asigna los valores residuales de predicción para el bloque actual a los valores de índice que indican entradas en la paleta para el bloque actual usados para representar los valores residuales de predicción para el bloque actual, y codifica los valores residuales de predicción usando los valores de índice. El descodificador de vídeo 30 puede obtener el bloque de valores de
20 índice a partir del flujo de bits, y determinar los valores residuales de predicción para el bloque actual basándose en los valores residuales de predicción correspondientes en la paleta que se identifican mediante los valores de índice. El descodificador de vídeo 30 puede entonces reconstruir los valores de píxel del bloque actual usando procedimientos de descodificación regulares basándose en los valores residuales de predicción y los valores de píxel de referencia codificados previamente.

30 En algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden realizar la codificación de vídeo basada en paleta con predicción de bloques de vídeo aplicando el modo de intra-predicción (es decir, la predicción solo usa información de píxel codificada previamente en la imagen actual). En otros casos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden aplicar el modo de inter-predicción (es decir, la predicción es a partir de píxeles en una imagen codificada previamente). En algunos casos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar los valores residuales de predicción para el bloque actual usando solo un subconjunto de procesos del modo de predicción para el modo de inter-predicción o bien el modo de intra-predicción.

40 En otro ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden no realizar ninguna predicción para el bloque actual. En este caso, el codificador de vídeo 20 en su lugar asigna los valores de píxel a índices de paleta, y codifica los índices usando codificación por entropía sin predicción. En un ejemplo adicional, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden realizar una modulación por codificación de impulsos diferenciales residuales (RDPCM) usando valores de píxel del bloque actual que se asignan a valores de índice de paleta. En este ejemplo, no se usa ninguna predicción de píxeles fuera del bloque actual, y se pueden usar predicciones horizontales o verticales para valores de índice de copiado de líneas dentro de la CU actual.

50 En algunos ejemplos, las técnicas para la codificación basada en paleta de datos de vídeo pueden usarse con una o más técnicas de codificación diferentes, tales como técnicas para la codificación inter o intra-predictiva. Por ejemplo, como se describe con más detalle a continuación, se puede configurar un codificador o descodificador, o un codificador-descodificador combinado (codec), para realizar codificación inter e intra-predictiva, así como codificación basada en paleta.

La FIG. 2 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo 20 que puede implementar las técnicas de la presente divulgación. La FIG. 2 se proporciona con fines explicativos y no debe considerarse limitativa de las técnicas ampliamente ejemplificadas y descritas en la presente divulgación. Con fines explicativos, la presente divulgación describe el codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de la presente divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

60 El codificador de vídeo 20 representa un ejemplo de un dispositivo que puede estar configurado para realizar técnicas para la codificación de vídeo basada en paleta de acuerdo con diversos ejemplos descritos en la presente divulgación. Por ejemplo, el codificador de vídeo 20 puede estar configurado para codificar selectivamente varios bloques de datos de vídeo, tales como CU o PU en codificación HEVC, usando codificación basada en paleta o codificación no basada en paleta. Los modos de codificación no basados en paleta pueden referirse a varios modos de codificación temporal inter-predictiva o modos de codificación espacial intra-predictiva, tales como los diversos modos de codificación especificados por el Borrador 10 de HEVC. El codificador de vídeo 20, en un ejemplo, puede
65

configurarse para generar una paleta que tiene entradas que indican valores de píxel. Asimismo, en este ejemplo, el codificador de vídeo 20 puede seleccionar valores de píxel en una paleta para representar valores de píxel de al menos algunas posiciones de un bloque de datos de vídeo. En este ejemplo, el codificador de vídeo 20 puede señalar información que asocia al menos algunas de las posiciones del bloque de datos de vídeo con entradas en la paleta correspondientes, respectivamente, a los valores de píxel seleccionados. El descodificador de vídeo 30 puede usar la información señalizada para descodificar datos de vídeo.

En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una memoria de datos de vídeo 98, una unidad de procesamiento de predicción 100, una unidad de generación de residuos 102, una unidad de procesamiento de transformación 104, una unidad de cuantización 106, una unidad de cuantización inversa 108, una unidad de procesamiento de transformación inversa 110, una unidad de reconstrucción 112, una unidad de filtro 114, una memoria intermedia de imágenes descodificadas 116 y una unidad de codificación por entropía 118. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de inter-predicción 120 y una unidad de procesamiento de intra-predicción 126. La unidad de procesamiento inter-predicción 120 incluye una unidad de estimación de movimiento y una unidad de compensación de movimiento (no mostrada). El codificador de vídeo 20 también incluye una unidad de codificación basada en paleta 122 configurada para llevar a cabo diversos aspectos de las técnicas de codificación basadas en paleta descritas en la presente divulgación. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales.

La memoria de datos de vídeo 98 puede almacenar datos de vídeo para codificarse mediante los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 98 pueden obtenerse, por ejemplo, del origen de vídeo 18. La memoria intermedia de imágenes descodificadas 116 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo mediante el codificador de vídeo 20, por ejemplo, en modos de intra o inter-codificación. La memoria de datos de vídeo 98 y la memoria intermedia de imágenes descodificadas 116 pueden estar formadas por cualquiera de diversos dispositivos de memoria, tales como memoria de acceso aleatorio dinámica (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 98 y la memoria intermedia de imágenes descodificadas 116 pueden proporcionarse mediante el mismo dispositivo de memoria o dispositivos de memoria independientes. En varios ejemplos, la memoria de datos de vídeo 98 puede estar en el chip con otros componentes del codificador de vídeo 20, o fuera del chip con respecto a esos componentes.

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 con bloques de árbol de codificación de luma (CTB) de tamaño igual y CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede realizar reparticiones de árbol cuádruple para dividir los CTB de la CTU en bloques progresivamente más pequeños. El bloque más pequeño puede ser bloques de codificación de CU. Por ejemplo, la unidad de procesamiento de predicción 100 puede repartir un CTB asociado con una CTU en cuatro sub-bloques de igual tamaño, repartir uno o más de los sub-bloques en cuatro sub-sub-bloques de igual tamaño, y así sucesivamente.

El codificador de vídeo 20 puede codificar 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 repartir los bloques de codificación asociados con la CU entre una o más PU de la CU. Así pues, cada PU puede estar asociada con un bloque de predicción de luma y los bloques de predicción de croma correspondientes. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir PU que tienen varios tamaños. Como se ha indicado anteriormente, el tamaño de una CU puede referirse al tamaño del bloque de codificación de luma de la CU y el tamaño de una PU puede referirse al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es $2N \times 2N$, 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 la intra-predicción y tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares para la inter-predicción. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden admitir reparticiones asimétricas para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para la inter-predicción.

La unidad de procesamiento inter-predicción 120 puede generar datos predictivos para una PU realizando una inter-predicción en cada PU de una CU. Los datos predictivos para la PU pueden incluir uno o más bloques de muestras predictivos de la PU e información de movimiento para la PU. La unidad de inter-predicción 121 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento I, todas las PU se intra-predicen. Por lo tanto, si la PU está en un fragmento I, la unidad de inter-predicción 121 no realiza inter-predicción en la PU. Así pues, para bloques codificados en modo I, el bloque predictivo se forma usando predicción espacial a partir de bloques contiguos codificados previamente dentro de la misma trama.

Si una PU está en un fragmento P, la unidad de estimación de movimiento de la unidad de procesamiento de inter-predicción 120 puede buscar las imágenes de referencia en una lista de imágenes de referencia (por ejemplo, "ListalmagRef0") para una región de referencia para la PU. La región de referencia para la PU puede ser una región, dentro de una imagen de referencia, que contiene bloques de muestras que se corresponden más estrechamente

con los bloques de muestras de la PU. La unidad de estimación de movimiento puede generar un índice de referencia que indica una posición en ListalmagRef0 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 con la región de referencia. Por ejemplo, el MV puede ser un vector bidimensional que proporciona un desplazamiento desde las coordenadas en la imagen descodificada actual a coordenadas en una imagen de referencia. La unidad de estimación de movimiento puede enviar 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 inter-predicción 120 puede generar los bloques de muestras predictivos de la PU basándose en muestras reales o interpoladas en la localización de referencia indicada por el vector de movimiento de la PU.

Si una PU está en un fragmento B, la unidad de estimación de movimiento puede realizar uni-predicción o bi-predicción para la PU. Para realizar uni-predicción para la PU, la unidad de estimación de movimiento puede buscar las imágenes de referencia de ListalmagRef0 o una segunda lista de imágenes de referencia ("ListalmagRef1") para una región de referencia para la PU. La unidad de estimación de movimiento puede enviar, como información de movimiento de la PU, un índice de referencia que indica una posición en ListalmagRef0 o ListalmagRef1 de la imagen de referencia que contiene la región de referencia, un MV que indica un desplazamiento espacial entre un bloque de muestras de la PU y una localización de referencia asociada con la región de referencia, y uno o más indicadores de dirección de predicción que indican si la imagen de referencia está en ListalmagRef0 o ListalmagRef1. La unidad de compensación de movimiento de la unidad de procesamiento de inter-predicción 120 puede generar los bloques de muestras 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.

Para realizar una inter-predicción bidireccional para una PU, la unidad de estimación de movimiento puede buscar las imágenes de referencia en ListalmagRef0 para una región de referencia para la PU y también puede buscar las imágenes de referencia en ListalmagRef1 para otra región de referencia para la PU. La unidad de estimación de movimiento puede generar índices de imágenes de referencia que indican posiciones en ListalmagRef0 y ListalmagRef1 de las imágenes de referencia que contienen las regiones de referencia. Además, la unidad de estimación de movimiento puede generar MV que indican desplazamientos espaciales entre la localización de referencia asociada con 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 de muestras 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.

De acuerdo con varios ejemplos de la presente divulgación, el codificador de vídeo 20 puede estar configurado para realizar una codificación basada en paleta. Con respecto a la estructura de HEVC, como un ejemplo, las técnicas de codificación basadas en paleta pueden configurarse para usarse como modo de CU. En otros ejemplos, las técnicas de codificación basadas en paleta pueden configurarse para usarse como un modo de PU en la estructura de HEVC. En consecuencia, todos los procesos divulgados descritos en el presente documento (a lo largo de la presente divulgación) en el contexto de un modo de CU pueden, de forma adicional o de forma alternativa, aplicarse a un modo de PU. Sin embargo, estos ejemplos basados en HEVC no deben considerarse una restricción o limitación de las técnicas de codificación basadas en paleta descritas en el presente documento, ya que dichas técnicas pueden aplicarse para trabajar independientemente o como parte de otros sistemas/normas existentes o aún no desarrollados. En estos casos, la unidad para la codificación de paleta pueden ser bloques cuadrados, bloques rectangulares o incluso regiones de forma no rectangular.

La unidad de codificación basada en paleta 122, por ejemplo, puede realizar una descodificación basada en paleta cuando se selecciona un modo de codificación basado en paleta, por ejemplo, para una CU o PU. Por ejemplo, la unidad de codificación basada en paleta 122 puede configurarse para generar una paleta que tiene entradas que indican valores de píxel, seleccionar valores de píxel en una paleta para representar valores de píxel de al menos algunas posiciones de un bloque de datos de vídeo, y señalar información que asocia al menos algunas de las posiciones del bloque de datos de vídeo con entradas en la paleta correspondientes, respectivamente, a los valores de píxel seleccionados. Aunque se describen diversas funciones como realizadas por la unidad de codificación basada en paleta 122, algunas o la totalidad de dichas funciones pueden realizarse mediante otras unidades de procesamiento, o una combinación de diferentes unidades de procesamiento.

La unidad de codificación basada en paleta 122 puede estar configurada para generar cualquiera de los diversos elementos sintácticos descritos en el presente documento. En consecuencia, el codificador de vídeo 20 puede estar configurado para codificar bloques de datos de vídeo usando modos de codificación basados en paleta como se describe en la presente divulgación. El codificador de vídeo 20 puede codificar selectivamente un bloque de datos de vídeo usando un modo de codificación de paleta, o codificar un bloque de datos de vídeo usando un modo diferente, por ejemplo, un modo de codificación HEVC inter-predictiva o intra-predictiva. El bloque de datos de vídeo puede ser, por ejemplo, una CU o PU generada de acuerdo con un proceso de codificación HEVC. Un codificador de vídeo 20 puede codificar algunos bloques con modos de predicción temporal inter-predictiva o codificación espacial intra-predictiva y descodificar otros bloques con el modo de codificación basado en paleta.

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

5 Para realizar la intra-predicción en una PU, la unidad de procesamiento de intra-predicción 126 puede usar múltiples modos de intra-predicción para generar múltiples conjuntos de datos predictivos para la PU. Cuando se usan algunos modos de intra-predicción para generar un conjunto de datos predictivos para la PU, la unidad de procesamiento de intra-predicción 126 puede extender valores de muestras de bloques de muestras de PU
10 contiguas en los bloques predictivos de la PU en direcciones asociadas con los modos de intra-predicción. Las PU contiguas pueden estar encima, encima y a la derecha, encima y a la izquierda, o a la izquierda de la PU, asumiendo un orden de codificación de izquierda a derecha, de arriba a abajo para PU, CU y CTU. La unidad de procesamiento de intra-predicción 126 puede usar varios números de modos de intra-predicción, por ejemplo, 33 modos de intra-predicción direccional. En algunos ejemplos, el número de modos de intra-predicción puede depender del tamaño de
15 la región asociada con la PU.

La unidad 100 de procesamiento de predicción puede seleccionar los datos predictivos para PU de una CU de entre los datos predictivos generados por la unidad de procesamiento de inter-predicción 120 para las PU o los datos predictivos generados por la unidad de procesamiento de intra-predicción 126 para las PU. En algunos ejemplos, la
20 unidad 100 de procesamiento de predicción selecciona los datos predictivos para las PU de la CU basándose en métricas de velocidad/distorsión de los conjuntos de datos predictivos. Los bloques de muestras predictivos de los datos predictivos seleccionados se pueden denominar en el presente documento bloques de muestras predictivos seleccionados.

25 La unidad de generación de residuos 102 puede generar, basándose en los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) de una CU y los bloques de muestras predictivos seleccionados (por ejemplo, bloques predictivos de luma, Cb y Cr) de las PU de la CU, bloques residuales (por ejemplo, bloques residuales de luma, Cb y Cr) de la CU. Por ejemplo, la unidad de generación de residuos 102 puede generar los bloques residuales de la CU de tal manera que cada muestra en los bloques residuales tiene un valor igual a una
30 diferencia entre una muestra en un bloque de codificación de la CU y una muestra correspondiente en un bloque de muestras predictivo seleccionado correspondiente de una PU de la CU.

La unidad de procesamiento de transformación 104 puede realizar reparticiones de árbol cuádruple para repartir los bloques residuales asociados a una CU en bloques de transformación asociados con TU de la CU. Así pues, en algunos ejemplos, una TU puede estar asociada con un bloque de transformación de luma y dos bloques de transformación de croma. Los tamaños y posiciones de los bloques de transformación de luma y croma de TU de una CU pueden o no basarse en los tamaños y posiciones de los bloques de predicción de las PU de la CU. Una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT) puede incluir nodos asociados con cada una de las regiones. Las TU de una CU pueden corresponder a nodos hoja del RQT.

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

50 La unidad de cuantización 106 puede cuantizar los coeficientes de transformación en un bloque de coeficientes. El proceso de cuantización puede reducir la profundidad de bits asociada con algunos o la totalidad de los coeficientes de transformación. Por ejemplo, un coeficiente de transformación de n bits puede redondearse por defecto a un coeficiente de transformación de m bits durante la cuantización, donde n es mayor que m . La unidad de cuantización 106 puede cuantizar un bloque de coeficientes asociado con una TU de una CU basándose en un valor del parámetro de cuantización (QP) asociado con la CU. El codificador de vídeo 20 puede ajustar el grado de cuantización aplicado a los bloques de coeficientes asociados con una CU ajustando el valor del QP asociado con la CU. La cuantización puede introducir pérdida de información, por lo que los coeficientes de transformación cuantizados pueden tener una precisión inferior a los originales.

60 La unidad de cuantización inversa 108 y la unidad de procesamiento de transformación inversa 110 pueden aplicar la cuantización inversa y transformaciones inversas a un bloque de coeficientes, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes. La unidad de reconstrucción 112 puede añadir el bloque residual reconstruido a muestras correspondientes de uno o más bloques de muestras predictivos generados por la unidad de procesamiento de predicción 100 para producir un bloque de transformación reconstruido asociado con una TU. Mediante la reconstrucción de los bloques de transformación para cada TU de una CU de esta manera, el
65 codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

La unidad de filtro 114 puede realizar una o más operaciones de desbloqueo para reducir los defectos de bloqueo en los bloques de codificación asociados con una CU. La memoria intermedia de imágenes descodificadas 116 puede almacenar los bloques de codificación reconstruidos después de que la unidad de filtro 114 realiza una o más operaciones de desbloqueo en los bloques de codificación reconstruidos. La unidad de procesamiento de inter-predicción 120 puede usar una imagen de referencia que contiene los bloques de codificación reconstruidos para realizar la inter-predicción en PU de otras imágenes. Además, la unidad de procesamiento de intra-predicción 126 puede usar bloques de codificación reconstruidos en la memoria intermedia de imágenes descodificadas 116 para realizar intra-predicción en otras PU en la misma imagen que la CU.

La unidad de codificación por entropía 118 puede recibir datos de otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 118 puede recibir bloques de coeficientes de la unidad de cuantización 106 y puede recibir elementos sintácticos de la unidad de procesamiento de predicción 100. La unidad de codificación por entropía 118 puede realizar una o más operaciones de codificación por entropía en los datos para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 118 puede realizar una operación CABAC, una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de repartición de intervalos de probabilidad (PIPE), una operación de codificación de Golomb exponencial, u otro tipo de operación de codificación por entropía en los datos. El codificador de vídeo 20 puede enviar un flujo de bits que incluye datos codificados por entropía generados por la unidad de codificación por entropía 118. Por ejemplo, el flujo de bits puede incluir datos que representan un RQT para una CU.

En algunos ejemplos, la codificación de residuos no se realiza con la codificación de paleta. En consecuencia, el codificador de vídeo 20 puede no realizar la transformación o cuantización cuando codifica usando un modo de codificación de paleta. Además, el codificador de vídeo 20 puede codificar por entropía los datos generados usando un modo de codificación de paleta independientemente de los datos residuales.

De acuerdo con una o más de las técnicas de la presente divulgación, el codificador de vídeo 20 y, específicamente la unidad de codificación basada en paleta 122, pueden realizar la codificación de vídeo basada en paleta de bloques de vídeo predichos. Como se ha descrito anteriormente, una paleta generada por el codificador de vídeo 20 puede codificarse explícitamente y enviarse al descodificador de vídeo 30, predecirse a partir de entradas de paleta anteriores, predecirse a partir de valores de píxel anteriores, o una combinación de los mismos.

En un ejemplo, la unidad de codificación basada en paleta 122 del codificador de vídeo 20 determina una o más entradas de paleta en una paleta predictiva que se copian a una paleta actual para un bloque actual de datos de vídeo, y determina un número de entradas de paleta nuevas que no están en la paleta predictora pero que se incluyen en la paleta actual. Basándose en esta información, el codificador de vídeo basado en paleta 20 calcula un tamaño de la paleta actual igual a la suma del número de las entradas de paleta copiadas y el número de las entradas de paleta nuevas, y genera la paleta actual del tamaño determinado incluyendo las entradas de paleta copiadas y las entradas de paleta nuevas. El codificador de vídeo 20 puede transmitir la información determinada relativa a las entradas de paleta copiadas y las entradas de paleta nuevas al descodificador de vídeo 30. Además, el codificador de vídeo 20 puede codificar y transmitir expresamente valores de píxel para las entradas de paleta nuevas al descodificador de vídeo 30. La unidad de codificación basada en paleta 122 del codificador de vídeo 20 puede entonces codificar el bloque actual determinando valores de índice para uno o más valores de píxel del bloque actual que identifican las entradas de paleta en la paleta actual usada para representar los valores de píxel del bloque actual.

Las técnicas descritas en la presente divulgación también pueden incluir técnicas para varias combinaciones de uno o más modos de codificación basados en paletas de señalización, paletas de transmisión, paletas de predicción, paletas de deducción, o mapas de codificación basados en paletas de transmisión y otros elementos sintácticos.

La FIG. 3 es un diagrama de bloques que ilustra un descodificador de vídeo de ejemplo 30 que está configurado para implementar las técnicas descritas en la presente divulgación. La FIG. 3 se proporciona con fines explicativos y no es limitativa en las técnicas ampliamente ejemplificadas y descritas en la presente divulgación. Con fines explicativos, la presente divulgación describe el descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de la presente divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

El descodificador de vídeo 30 representa un ejemplo de un dispositivo que puede estar configurado para realizar técnicas para la codificación de vídeo basada en paleta de acuerdo con diversos ejemplos descritos en la presente divulgación. Por ejemplo, el descodificador de vídeo 30 puede estar configurado para descodificar selectivamente diversos bloques de datos de vídeo, tales como CU o PU en la codificación HEVC, usando codificación basada en paleta o bien codificación no basada en paleta. Los modos de codificación no basados en paleta pueden referirse a varios modos de codificación temporal inter-predictiva o modos de codificación espacial intra-predictiva, tales como los diversos modos de codificación especificados por el Borrador 10 de HEVC. En un ejemplo, el descodificador de

vídeo 30 puede estar configurado para generar una paleta que tiene entradas que indican valores de píxel. Asimismo, en este ejemplo, el descodificador de vídeo 30 puede recibir información que asocia al menos algunas posiciones de un bloque de datos de vídeo con entradas en la paleta. En este ejemplo, el descodificador de vídeo 30 puede seleccionar valores de píxel en la paleta basándose en la información y reconstruir valores de píxel del bloque basándose en los valores de píxel seleccionados.

En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una memoria de datos de vídeo 148, una unidad de descodificación por entropía 150, una unidad de procesamiento de predicción 152, una unidad de cuantización inversa 154, una unidad de procesamiento de transformación inversa 156, una unidad de reconstrucción 158, una unidad de filtro 160 y una memoria intermedia de imágenes descodificadas 162. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 164 y la unidad de procesamiento de intra-predicción 166. El descodificador de vídeo 30 también incluye una unidad de descodificación basada en paleta 165 configurada para llevar a cabo diversos aspectos de las técnicas de codificación basadas en paleta descritas en la presente divulgación. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

La memoria de datos de vídeo 148 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificados, para ser descodificarse mediante los componentes del descodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 148 pueden obtenerse, por ejemplo, a partir de un medio legible por ordenador 16, por ejemplo a partir de un origen de vídeo local, tal como una cámara, a través de una comunicación de datos de vídeo por cable o inalámbrica, o mediante el acceso a medios de almacenamiento de datos físicos. La memoria de datos de vídeo 148 puede formar una memoria intermedia de imágenes codificadas (CPB) que almacena datos de vídeo codificados de un flujo de bits de vídeo codificados. La memoria intermedia de imágenes descodificadas 162 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la descodificación de datos de vídeo mediante el descodificador de vídeo 30, por ejemplo, en modos de intra o inter-codificación. La memoria de datos de vídeo 148 y la memoria intermedia de imágenes descodificadas 162 pueden estar formadas por cualquiera de diversos dispositivos de memoria, tales como memoria de acceso aleatorio dinámica (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 148 y la memoria intermedia de imágenes descodificadas 162 pueden proporcionarse mediante el mismo dispositivo de memoria o dispositivos de memoria independientes. En varios ejemplos, la memoria de datos de vídeo 148 puede estar en el chip con otros componentes del descodificador de vídeo 30, o fuera del chip con respecto a esos componentes.

La memoria 148 de datos de vídeo, es decir, una CPB, puede recibir y almacenar datos de vídeo codificados (por ejemplo, unidades NAL) de un flujo de bits. La unidad 150 de descodificación por entropía puede recibir datos de vídeo codificados (por ejemplo, unidades NAL) de la memoria de datos de vídeo 148 y puede analizar las unidades NAL para descodificar elementos sintácticos. La unidad de descodificación por entropía 150 puede descodificar por entropía elementos sintácticos codificados por entropía en las unidades NAL. La unidad de procesamiento de predicción 152, la unidad de cuantización inversa 154, la unidad de procesamiento de transformación inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 160 pueden generar datos de vídeo descodificados basándose en los elementos sintácticos obtenidos (por ejemplo, extraídos) del flujo de bits.

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

Además de descodificar elementos sintácticos a partir del flujo de bits, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU no repartida. Para realizar la operación de reconstrucción en una CU no repartida, 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.

Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantización inversa 154 puede cuantizar inversamente, es decir, descuantizar, bloques de coeficientes asociados con la TU. La unidad de cuantización inversa 154 puede utilizar un valor del QP asociado con la CU de la TU para determinar un grado de cuantización y, asimismo, un grado de cuantización inversa a aplicar en la unidad de cuantización inversa 154. Es decir, la relación de compresión, es decir, la relación entre el número de bits usados para representar la secuencia original y la comprimida, puede controlarse ajustando el valor del QP usado al cuantizar los coeficientes de transformación. La relación de compresión también puede depender del procedimiento de codificación por entropía empleado.

Después de que la unidad de cuantización inversa 154 cuantiza de forma inversa un bloque de coeficientes, la

unidad de procesamiento de transformación inversa 156 puede aplicar una o más transformaciones inversas al bloque de coeficientes con el fin de generar un bloque residual asociado con la TU. Por ejemplo, la unidad de procesamiento de transformación inversa 156 puede aplicar una DCT inversa, una transformación entera inversa, una transformación de Karhunen-Loeve (KLT) inversa, una transformación de rotación inversa, una transformación direccional inversa, u otra transformación inversa al bloque de coeficientes.

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

La unidad de procesamiento de predicción 152 puede construir una primera lista de imágenes de referencia (ListalmagRef0) y una segunda lista de imágenes de referencia (ListalmagRef1) basándose en elementos sintácticos extraídos del flujo de bits. Asimismo, si se codifica una PU usando inter-predicción, 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.

La unidad de reconstrucción 158 puede usar los bloques de transformación (por ejemplo, bloques de transformación de luma, Cb y Cr) asociados con las TU de una CU y los bloques predictivos (por ejemplo, bloques de luma, Cb y Cr) de las PU de la CU, es decir, datos de intra-predicción o datos de inter-predicción, 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 añadir muestras de los bloques de transformación (por ejemplo, bloques de transformación de luma, Cb y Cr) a muestras correspondientes de los bloques predictivos (por ejemplo, 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.

La unidad de filtro 160 puede realizar una operación de desbloqueo para reducir los defectos de bloqueo asociados con los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) de la CU. El descodificador de vídeo 30 puede almacenar los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) de la CU en la memoria intermedia de imágenes descodificadas 162. La memoria intermedia de imágenes descodificadas 162 puede proporcionar imágenes de referencia para la posterior compensación de movimiento, intra-predicción 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, bloques de luma, Cb y Cr) en la memoria intermedia de imágenes descodificadas 162, operaciones de intra-predicción o inter-predicción en las PU de otras CU. De esta manera, el descodificador de vídeo 30 puede extraer, a partir del flujo de bits, niveles de coeficientes de transformación de un bloque de coeficientes significativos, cuantizar de forma inversa los niveles de coeficientes de transformación, aplicar una transformación a los niveles de coeficientes de transformación para generar un bloque de transformación, generar, basándose al menos en parte en el bloque de transformación, un bloque de codificación, y enviar el bloque de codificación para su visualización.

De acuerdo con diversos ejemplos de la presente divulgación, el descodificador de vídeo 30 puede estar configurado para realizar una codificación basada en paleta. La unidad de descodificación basada en paleta 165, por ejemplo, puede realizar una descodificación basada en paleta cuando se selecciona un modo de descodificación basado en paleta, por ejemplo, para una CU o PU. Por ejemplo, la unidad de descodificación basada en paletas 165 puede estar configurada para generar una paleta que tiene entradas que indican valores de píxel. Asimismo, en este ejemplo, la unidad de descodificación basada en paletas 165 puede recibir información que asocia al menos algunas posiciones de un bloque de datos de vídeo con entradas en la paleta. En este ejemplo, la unidad de descodificación basada en paletas 165 puede seleccionar valores de píxel en la paleta basándose en la información. De forma adicional, en este ejemplo, la unidad de descodificación basada en paletas 165 puede reconstruir valores de píxel del bloque basándose en los valores de píxel seleccionados. Aunque se describen varias funciones como realizadas por la unidad de descodificación basada en paletas 165, algunas o la totalidad de estas funciones pueden realizarse mediante otras unidades de procesamiento, o una combinación de diferentes unidades de procesamiento.

La unidad de descodificación basada en paletas 165 puede recibir información del modo de codificación de paleta, y realizar las operaciones anteriores cuando la información del modo de codificación de paleta indica que el modo de codificación de paleta se aplica al bloque. Cuando la información del modo de codificación de paleta indica que el modo de codificación de paleta no se aplica al bloque, o cuando otra información del modo indica el uso de un modo diferente, la unidad de descodificación basada en paletas 165 descodifica el bloque de datos de vídeo usando un modo de codificación no basado en paletas, por ejemplo, un modo de codificación HEVC inter-predictiva o intra-predictiva, cuando la información del modo de codificación de paleta indica que el modo de codificación de paleta no se aplica al bloque. El bloque de datos de vídeo puede ser, por ejemplo, una CU o PU generada de acuerdo con un

proceso de codificación HEVC. Un descodificador de vídeo 30 puede descodificar algunos bloques con modos de predicción temporal inter-predictiva o de codificación espacial intra-predictiva y descodificar otros bloques con el modo de codificación basado en paletas. El modo de codificación basado en paletas puede comprender uno de una pluralidad de diferentes modos de codificación basados en paletas, o puede haber un único modo de codificación basado en paletas.

De acuerdo con una o más de las técnicas de la presente divulgación, el descodificador de vídeo 30 y, específicamente la unidad de descodificación basada en paletas 165, pueden realizar la descodificación de vídeo basada en paletas de bloques de vídeo predichos. Como se ha descrito anteriormente, una paleta generada por el descodificador de vídeo 30 puede codificarse explícitamente mediante el codificador de vídeo 20, predecirse a partir de entradas de paleta anteriores, predecirse a partir de valores de píxel anteriores, o una combinación de los mismos.

En un ejemplo, la unidad de descodificación basada en paletas 165 del descodificador de vídeo 30 determina una o más entradas de paleta en una paleta predictiva que se copian a una paleta actual para un bloque actual de datos de vídeo, y determina un número de entradas de paleta nuevas que no están en la paleta predictora pero que se incluyen en la paleta actual. El descodificador de vídeo 30 puede recibir la información relativa a las entradas de paleta copiadas y las entradas de paleta nuevas del codificador de vídeo 20. Además, el descodificador de vídeo 30 puede recibir valores de píxel codificados explícitamente para las entradas de paleta nuevas transmitidas desde el codificador de vídeo 20. Basándose en esta información, la unidad de descodificación basada en paletas 165 calcula un tamaño de la paleta actual igual a la suma del número de las entradas de paleta copiadas y el número de las entradas de paleta nuevas, y genera la paleta actual del tamaño determinado incluyendo las entradas de paleta copiadas y las entradas de paleta nuevas. La unidad de descodificación basada en paletas 165 del descodificador de vídeo 30 puede entonces descodificar el bloque actual determinando valores de índice para uno o más valores de píxel del bloque actual que identifican las entradas de paleta en la paleta actual usadas para representar los valores de píxel del bloque actual.

Las técnicas descritas en la presente divulgación también pueden incluir técnicas para varias combinaciones de uno o más modos de codificación basados en paletas de señalización, paletas de transmisión, paletas de predicción, paletas de deducción, o mapas de codificación basados en paletas de transmisión y otros elementos sintácticos.

Como se ha descrito anteriormente, en algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden realizar la codificación basada en paletas de bloques de vídeo predichos. En un ejemplo, el codificador de vídeo 20 obtiene en primer lugar una paleta para un bloque actual basándose en los valores de píxel en el bloque actual y, a continuación, asigna los valores de píxel en el bloque actual a índices de paleta para la codificación. La asignación puede ser de uno a uno (es decir, para la codificación sin pérdidas) o de varios a uno (es decir, para la codificación con pérdidas). El codificador de vídeo 20 también puede asignar valores de píxel de referencia en un bloque codificado previamente que pueden usarse para predecir los valores de píxel en el bloque actual. Una vez que el codificador de vídeo 20 asigna los valores de píxel del bloque actual a índices de paleta, el codificador de vídeo 20 codifica el bloque actual usando procedimientos de codificación regulares, por ejemplo, intra-codificación regular en la norma HEVC.

En el ejemplo anterior, el bloque actual con índices de paleta se trata como si fuera un bloque original con valores de píxel. Del mismo modo, los índices de paleta de los píxeles de referencia se utilizan para realizar una intra-predicción regular en el bloque actual con índices de paleta. El codificador de vídeo 20 puede transmitir el error de predicción o los valores residuales al descodificador de vídeo 30. En algunos casos, el error de predicción o los valores residuales pueden transformarse, cuantizarse y codificarse por entropía en el flujo de bits. En otros casos, también es posible que la transformación y la cuantización estén deshabilitadas para el modo de codificación de paleta. Después de codificar el bloque actual, el codificador de vídeo 20 puede convertir los índices de los píxeles de referencia, los píxeles de predicción, y/o los valores residuales en los valores de píxel para la reconstrucción del bloque actual y la predicción normal de bloques futuros. El descodificador de vídeo 30 puede obtener los valores residuales codificados para el bloque actual a partir del flujo de bits. Asimismo, el descodificador de vídeo 30 puede descodificar el bloque actual usando un procedimiento de descodificación regular para obtener el bloque actual con índices de paleta. El descodificador de vídeo 30 puede entonces determinar los valores de píxel del bloque actual basándose en los valores de píxel en la paleta que están asociados con los índices de paleta.

En otro ejemplo, el codificador de vídeo 20 puede generar una paleta para un bloque actual. La paleta puede incluir entradas que indican valores residuales de predicción para el bloque dado. Los valores residuales de predicción para el bloque dado se pueden generar usando cualquier modo de predicción, por ejemplo, inter-predicción o intra-predicción regular en la norma HEVC. Los valores residuales de predicción para el bloque dado pueden ser valores de píxel residuales (posiblemente cuantizados) o valores de coeficientes de transformación residuales (posiblemente cuantizados). En este ejemplo, el codificador de vídeo 20 asigna los valores residuales de predicción para el bloque actual a valores de índice que indican entradas en la paleta para el bloque actual que se usan para representar los valores residuales de predicción para el bloque actual. En este ejemplo, el codificador de vídeo 20 puede codificar los valores de índice para una o más posiciones en el bloque actual, donde los valores de índice indican las entradas en la paleta para el bloque actual que especifican los valores residuales de predicción para el bloque actual. El

descodificador de vídeo 30 puede obtener el bloque codificado de valores de índice a partir del flujo de bits, y determinar los valores residuales de predicción para el bloque actual basándose en los valores residuales de predicción correspondientes en la paleta identificados mediante los valores de índice. El descodificador de vídeo 30 puede entonces reconstruir los valores de píxel del bloque actual usando procedimientos de descodificación regulares basándose en los valores residuales de predicción y los valores de píxel de referencia codificados previamente.

En algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden realizar la codificación de vídeo basada en paletas con predicción de bloques de vídeo aplicando el modo de intra-predicción (es decir, la predicción solo usa información de píxel codificada previamente en la imagen actual). En otros ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden aplicar el modo de inter-predicción (es decir, la predicción es a partir de píxeles en imágenes codificadas previamente). En un ejemplo, los valores residuales de predicción para el bloque actual pueden ser valores de píxel residuales para el bloque actual calculados a partir de los valores de píxel del bloque actual y los valores de píxel de referencia codificados previamente. Los valores de píxel residuales se pueden cuantizar. En otro ejemplo, los valores residuales de predicción para el bloque actual pueden ser valores de coeficientes de transformación residuales para el bloque actual calculados a partir de los valores de píxel del bloque actual y los valores de píxel de referencia codificados previamente, y posteriormente transformados y posiblemente cuantizados.

En algunos casos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar los valores residuales de predicción para el bloque actual usando solo un subconjunto de procesos del modo de predicción para el modo de inter-predicción o bien el modo de intra-predicción. Por ejemplo, en el caso del modo de intra-predicción, los procesos de predicción DC, horizontal y/o vertical pueden estar habilitados, pero otros procesos del modo de intra-predicción pueden estar deshabilitados. Los procesos deshabilitados pueden incluir el filtrado en el modo de intra-predicción, por ejemplo, uno o más de intra-suavizado dependiente del modo (MDIS), interpolación bilineal 1/32-pel, filtrado de bordes y/o filtrado DC (puede encontrarse una introducción preliminar en la Solicitud provisional de EE.UU. n.º 61/890 844, presentada el 14 de octubre de 2013, titulada "Adaptive Filter Control for Intra Prediction in Video Coding" ("Control de filtrado adaptativo para intra-predicción en codificación de vídeo"), número de referencia del solicitante 1212-671USP3/134960P3), está deshabilitado en este ejemplo. Como un ejemplo adicional, en el caso del modo de inter-predicción, el proceso de promediado de píxeles, por ejemplo, una o más de la predicción ponderada, la bi-predicción, o la interpolación sub-pel, puede estar deshabilitado.

En otro ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden no realizar ninguna predicción para el bloque actual. En este caso, el codificador de vídeo 20 en su lugar asigna los valores de píxel a índices de paleta, y codifica los índices usando codificación por entropía sin predicción. En un ejemplo adicional, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden realizar la modulación por codificación de impulsos diferenciales residuales (RDPCM) usando valores de píxel del bloque actual que se asignan a valores de índice de paleta. En este caso, no se usa ninguna predicción a partir de los píxeles fuera del bloque actual, y se puede usar predicción horizontal o vertical para los valores de índice de copiado de líneas dentro de la CU actual. Por ejemplo, cuando se usa la predicción vertical, las localizaciones de la primera fila del bloque actual no se predicen, y las localizaciones en las filas posteriores pueden predecirse usando valores en las filas anteriores, por ejemplo, los valores en la fila i ($i > 0$) iguales a $x(i, j)$ se predicen usando $x(i-1, j)$. Cuando se usa la predicción horizontal, las localizaciones en la primera columna del bloque actual no se predicen, y las localizaciones en las columnas posteriores se pueden predecir usando valores en las columnas anteriores.

En algunos ejemplos, las técnicas para la codificación basada en paletas de datos de vídeo pueden usarse con una o más técnicas de codificación diferentes, tales como técnicas para codificación inter o intra-predictiva. Por ejemplo, como se describe con más detalle a continuación, un codificador o descodificador, o un codificador-descodificador combinado (codec), se puede configurar para realizar codificación inter e intra-predictiva, así como codificación basada en paletas.

La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de determinación de una paleta para codificación de datos de vídeo, coherente con técnicas de la presente divulgación. El ejemplo de la FIG. 4 incluye una imagen 178 que tiene una primera unidad de codificación (CU) 180 que está asociada con un primer conjunto de paletas (es decir, primeras paletas 184) y una segunda CU 188 que está asociada con un segundo conjunto de paletas (es decir, segundas paletas 192). Como se describe con mayor detalle a continuación y de acuerdo con una o más de las técnicas de la presente divulgación, las segundas paletas 192 se basan en las primeras paletas 184. La imagen 178 también incluye el bloque 196 codificado con un modo de codificación de intra-predicción y el bloque 200 que está codificado con un modo de codificación de inter-predicción.

Las técnicas de la FIG. 4 se describen en el contexto del codificador de vídeo 20 (FIG. 1 y FIG. 2) y el descodificador de vídeo 30 (FIG. 1 y FIG. 3) y con respecto a la norma de codificación de vídeo HEVC con fines explicativos. Con respecto a la estructura de HEVC, como un ejemplo, las técnicas de codificación basada en paletas pueden configurarse para usarse como un modo de CU. En otros ejemplos, las técnicas de codificación basada en paletas pueden configurarse para usarse como un modo de PU o un modo de TU en la estructura de HEVC. En consecuencia, todos los procesos divulgados siguientes descritos en el contexto de un modo de CU pueden, de

forma adicional o de forma alternativa, aplicarse a una PU o una TU. Sin embargo, debe entenderse que las técnicas de la presente divulgación no están limitadas de esta manera, y pueden aplicarse mediante otros procesadores y/o dispositivos de codificación de vídeo en otros procesos y/o normas de codificación de vídeo.

En general, una paleta se refiere a un cierto número de valores de píxel que son dominantes y/o representativos para una CU que se está codificando actualmente (por ejemplo, la CU 188 en el ejemplo de la FIG. 4). Las primeras paletas 184 y segundas paletas 192 se muestran incluyendo múltiples paletas. En algunos ejemplos, de acuerdo con aspectos de la presente 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 de color de una CU. Por ejemplo, el codificador de vídeo 20 puede codificar una paleta para un componente de luma (Y) de una CU, otra paleta para un componente de croma (U) de la CU, y otra paleta más para el componente de croma (V) de la CU. En este ejemplo, las entradas de paleta Y pueden representar valores Y de píxeles de la CU, las entradas de paleta U pueden representar valores U de píxeles de la CU y las entradas de paleta V pueden representar valores V de píxeles de la CU.

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, incluyendo 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 tienen múltiples paletas individuales es simplemente un ejemplo y no pretende ser limitativa.

En el ejemplo de la FIG. 4, cada una de las primeras paletas 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 el valor de píxel A, el valor de píxel B y el valor de píxel C, respectivamente. Debe observarse que cada una de las primeras paletas 184 no incluye realmente los índices y cabeceras de columna, sino que solo incluye los valores de píxel A, B y C y los índices se usan para identificar las entradas en la paleta.

Como se describe en el presente documento, en lugar de codificar los valores de píxel reales de la primera CU 180, un codificador de vídeo (tal como el codificador de vídeo 20 o el descodificador de vídeo 30) puede usar codificación basada en paletas 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 con 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 un flujo de bits y puede reconstruir los valores de píxel usando los valores de índice y una o más de las primeras paletas 184. En otras palabras, para cada valor de índice respectivo para un bloque, el descodificador de vídeo 30 puede determinar una entrada en una de las primeras paletas 184. El descodificador de vídeo 30 puede sustituir el valor de índice respectivo en el bloque por el valor de píxel especificado por la entrada determinada en la paleta. El codificador de vídeo 20 puede transmitir las primeras paletas 184 en un flujo de bits de datos de vídeo codificados para su uso mediante el descodificador de vídeo 30 en la descodificación basada en paletas. En general, una o más paletas pueden transmitirse para cada CU o pueden compartirse entre diferentes CU.

De acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden determinar segundas paletas 192 basándose en primeras paletas 184. Por ejemplo, el codificador de vídeo 20 puede codificar un indicador_paleta_pred para cada CU (incluyendo, como un ejemplo, la segunda CU 188) para indicar si la paleta para la CU se predice a partir de una o más paletas asociadas con una o más CU, tal como CU contiguas (espacialmente o basándose en el orden de exploración) o las muestras más frecuentes de un entorno causal. Por ejemplo, cuando el valor de dicho indicador es igual a uno, el descodificador de vídeo 30 puede determinar que las segundas paletas 192 para la segunda CU 188 se predicen a partir de una o más paletas ya descodificadas y por lo tanto no se incluyen nuevas paletas para la segunda CU 188 en un flujo de bits que contiene el indicador_paleta_pred. Cuando dicho indicador es igual a cero, el descodificador de vídeo 30 puede determinar que las paletas 192 para la segunda CU 188 están incluidas en el flujo de bits como una nueva paleta. En algunos ejemplos, indicador_paleta_pred puede codificarse por separado para cada componente de color diferente de una CU (por ejemplo, tres indicadores, uno para Y, uno para U y uno para V, para una CU en vídeo YUV). En otros ejemplos, puede codificarse un único indicador_paleta_pred para todos los componentes de color de una CU.

En el ejemplo anterior, el indicador_paleta_pred se señala por CU para indicar si se predice alguna de las entradas de la paleta para el bloque actual. Esto significa que las segundas paletas 192 son idénticas a las primeras paletas 184 y no se señala ninguna información adicional. En otros ejemplos, uno o más elementos sintácticos pueden señalizarse en una base por entrada. Es decir, se puede señalar un indicador para cada entrada de un predictor de paleta para indicar si esa entrada está presente en la paleta actual. Como se ha observado anteriormente, si no se predice una entrada de paleta, la entrada de paleta puede señalizarse explícitamente. En otros ejemplos, estos dos procedimientos podrían combinarse. Por ejemplo, en primer lugar se señala el indicador_paleta_pred. Si el indicador es 0, se puede señalar un indicador de predicción por entrada. Además, se puede señalar el número de entradas nuevas y sus valores explícitos.

Quando se determinan segundas paletas 192 con respecto a primeras paletas 184 (por ejemplo, indicador_paleta_pred es igual a uno), el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden localizar uno o más bloques a partir de los que se determinan las paletas predictivas, en este ejemplo las primeras paletas 184. Las paletas predictivas pueden estar asociadas con una o más CU contiguas de la CU que se está codificando actualmente (por ejemplo, tal como las CU contiguas (espacialmente o basándose en el orden de exploración) o las muestras más frecuentes de un entorno causal), es decir, la segunda CU 188. Las paletas de una o más CU contiguas pueden estar asociadas con una paleta predictiva. En algunos ejemplos, tal como el ejemplo ilustrado en la FIG. 4, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden localizar una CU contigua izquierda, la primera CU 180, cuando se determina una paleta predictiva para la segunda CU 188. En otros ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden localizar una o más CU en otras posiciones con respecto a la segunda CU 188, tal como una CU superior, la CU 196. En otro ejemplo, la paleta para la última CU en el orden de exploración que usó el modo de paleta se puede usar como una paleta predictiva.

El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar una CU para la predicción de paletas basándose en una jerarquía. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden identificar inicialmente la CU contigua izquierda, la primera CU 180, para la predicción de paletas. Si la CU contigua izquierda no está disponible para la predicción (por ejemplo, la CU contigua izquierda está codificada con un modo distinto de un modo de codificación basado en paletas, tal como un modo de intra-predicción o un modo de intra-predicción, o está localizada en el borde del extremo izquierdo de una imagen o fragmento) el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden identificar la CU contigua superior, la CU 196. El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden continuar buscando una CU disponible de acuerdo con un orden predeterminado de localizaciones hasta localizar una CU que tiene una paleta disponible para la predicción de paletas. En algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar una paleta predictiva basándose en múltiples bloques y/o muestras reconstruidas de un bloque contiguo.

Aunque el ejemplo de la FIG. 4 ilustra las primeras paletas 184 como paletas predictivas a partir de una única CU (es decir, la primera CU 180), en otros ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden localizar paletas para la predicción a partir de una combinación de CU contiguas. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden aplicar una o más fórmulas, funciones, reglas o similares para generar una paleta predictiva basándose en paletas de una o una combinación de una pluralidad de CU contiguas (espacialmente o en orden de exploración).

En otros ejemplos adicionales, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden construir una lista de candidatos que incluye un cierto número de candidatos potenciales para la predicción de paletas. En dichos ejemplos, el codificador de vídeo 20 puede codificar un índice en la lista de candidatos para indicar la CU candidata en la lista a partir de la que se selecciona la CU actual usada para la predicción de paletas (por ejemplo, copia la paleta). El descodificador de vídeo 30 puede construir la lista de candidatos de la misma manera, descodificar el índice y usar el índice descodificado para seleccionar la paleta de la CU correspondiente para su uso con la CU actual. En otro ejemplo, la paleta de la CU candidata indicada en la lista puede usarse como una paleta predictiva para la predicción por entrada de una paleta actual para la CU actual.

En un ejemplo con fines ilustrativos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden construir una lista de candidatos que incluye una CU que está situada por encima de la CU que se está codificando actualmente y una CU que está situada a la izquierda de la CU que se está codificando actualmente. En este ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos para indicar la selección de candidatos. Por ejemplo, el codificador de vídeo 20 puede codificar un indicador que tiene un valor de cero para indicar que la paleta para la CU actual se copia desde la CU situada a la izquierda de la CU actual. El codificador de vídeo 20 puede codificar el indicador que tiene un valor de uno para indicar que la paleta para la CU actual se copia desde la CU situada encima de la CU actual. El descodificador de vídeo 30 descodifica el indicador y selecciona la CU apropiada para la predicción de paletas. En otro ejemplo, el indicador puede indicar si la paleta de la CU contigua superior o izquierda se usa como una paleta predictiva. Entonces, para cada entrada en la paleta predictiva, puede indicarse si esa entrada se usa en la paleta para la CU actual.

En otros ejemplos adicionales, el codificador de vídeo 20 y/o el descodificador de vídeo 30 determinan la paleta para la CU que se está codificando actualmente basándose en la frecuencia con la que ocurren valores de muestras incluidos en una o más paletas diferentes en una o más CU contiguas. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden monitorizar los colores asociados con los valores de índice usados con más frecuencia durante la codificación de un número predeterminado de CU. El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden incluir los colores usados con más frecuencia en la paleta para la CU que se está codificando actualmente.

Como se ha observado anteriormente, en algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo pueden copiar una paleta completa desde una CU contigua para codificar una CU actual. De forma adicional o de forma alternativa, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden realizar una predicción de paletas basada en entrada. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos para cada entrada de una paleta indicando si las entradas respectivas se predicen basándose en una paleta

predictiva (por ejemplo, una paleta de otra CU). En este ejemplo, el codificador de vídeo 20 puede codificar un indicador que tiene un valor de uno para una entrada dada cuando la entrada es un valor predicho de una paleta predictiva (por ejemplo, una entrada correspondiente de una paleta asociada con una CU contigua). El codificador de vídeo 20 puede codificar un indicador que tiene un valor de cero para una entrada particular para indicar que la entrada particular no se predice a partir de una paleta de otra CU. En este ejemplo, el codificador de vídeo 20 también puede codificar datos adicionales que indican el valor de la entrada de paleta no predicha.

La presente divulgación describe varias técnicas alternativas para predecir una paleta para una CU actual. En un ejemplo, una paleta predictiva que incluye entradas de paleta de una o más CU contiguas codificadas previamente incluye un número de entradas, N. En este caso, el codificador de vídeo 20 transmite en primer lugar un vector binario, V, que tiene el mismo tamaño que la paleta predictiva, es decir, el tamaño N, al descodificador de vídeo 30. Cada entrada en el vector binario indica si la entrada correspondiente en la paleta predictiva se reutilizará o copiará en la paleta para la CU actual. Por ejemplo, $V(i) = 1$ significa que la i-ésima entrada en la paleta predictiva para la CU contigua se reutilizará o copiará en la paleta para la CU actual, que puede tener un índice diferente en la CU actual.

Además, el codificador de vídeo 20 puede transmitir un número, M, que indica cuántas entradas de paleta nuevas están incluidas en la paleta para la CU actual y, a continuación, transmite un valor de píxel para cada una de las entradas de paleta nuevas al descodificador de vídeo 30. En este ejemplo, el tamaño final de la paleta para la CU actual puede obtenerse como igual a $M + S$, donde S es el número de entradas en la paleta predictiva que pueden reutilizarse o copiarse en la paleta para la CU actual (es decir, $V(i) = 1$). Para generar la paleta para la CU actual, el descodificador de vídeo 30 puede combinar las entradas de paleta nuevas transmitidas y las entradas de paleta copiadas reutilizadas de la paleta predictiva. En algunos casos, la combinación puede basarse en los valores de píxel, de tal manera que las entradas en la paleta para la CU actual puedan aumentar (o disminuir) con el índice de paleta. En otros casos, la combinación puede ser una concatenación de los dos conjuntos de entradas, es decir, las entradas de paleta nuevas y las entradas de paleta copiadas.

En otro ejemplo, el codificador de vídeo 20 transmite en primer lugar una indicación de un tamaño de una paleta, N, para un CU actual a un descodificador de vídeo 30. El codificador de vídeo 20 transmite entonces un vector, V, que tiene el mismo tamaño que la paleta para la CU actual, es decir, tamaño N, al descodificador de vídeo 30. Cada entrada en el vector indica si la entrada correspondiente en la paleta para la CU actual se transmite explícitamente mediante el codificador de vídeo 20 o se copia desde una paleta predictiva. Por ejemplo, $V(i) = 1$ significa que el codificador de vídeo 20 transmite la i-ésima entrada en la paleta al descodificador de vídeo 30, y $V(i) = 0$ significa que la i-ésima entrada en la paleta se copia desde la paleta predictiva. Para las entradas que se copian desde la paleta predictiva (es decir, $V(i) = 0$), el codificador de vídeo 20 puede usar diferentes procedimientos para señalar qué entrada en la paleta predictiva se usa en la paleta para la CU actual. En algunos casos, el codificador de vídeo 20 puede señalar el índice de paleta de la entrada a copiar desde la paleta predictiva a la paleta para la CU actual. En otros casos, el codificador de vídeo 20 puede señalar un desplazamiento del índice, que es la diferencia entre el índice en la paleta para la CU actual y el índice en la paleta predictiva.

En los dos ejemplos anteriores, la una o más CU contiguas codificadas previamente usadas para generar la paleta predictiva usada para la predicción de la paleta para la CU actual pueden ser una CU contigua superior (es decir, superior) o una CU contigua izquierda con respecto a la CU actual. En algunos ejemplos, puede construirse una lista de candidatos de CU contiguas, y el codificador de vídeo 20 transmite un índice para indicar qué CU contiguas candidatas y paletas asociadas se usan para la predicción de paletas para la CU actual. Para ciertas CU, por ejemplo, CU que se colocan al comienzo de un fragmento o en otros límites de fragmento o CU del extremo izquierdo en el fragmento o una imagen de datos de vídeo, la predicción de paletas puede estar deshabilitada.

En un ejemplo adicional, el codificador de vídeo 20 transmite una indicación de un número de entradas incluidas en una paleta para una CU actual al descodificador de vídeo 30. A continuación, para cada una de las entradas de paleta, el codificador de vídeo 20 transmite un indicador u otro elemento sintáctico para indicar si la entrada de paleta se transmite explícitamente mediante el codificador de vídeo 20 o si se obtiene a partir de un píxel reconstruido previamente. Por ejemplo, un indicador de un bit establecido igual a 1 puede significar que el codificador de vídeo 20 envía explícitamente la entrada de paleta, y el indicador de un bit establecido igual a 0 puede significar que la entrada de paleta se obtiene a partir de un píxel reconstruido previamente. Para cada una de las entradas de paleta que se obtienen a partir de un píxel reconstruido previamente, el codificador de vídeo 20 transmite otra indicación relativa a una localización de píxel del píxel reconstruido en la CU actual o una CU contigua que corresponde a la entrada de paleta. En algunos casos, la indicación de la localización de los píxeles reconstruidos puede ser un vector de desplazamiento relativo a la posición superior izquierda de la CU actual. En otros casos, la indicación de la localización de los píxeles reconstruidos puede ser un índice en una lista de píxeles reconstruidos que se puede usar para especificar la entrada de paleta para la CU actual. Por ejemplo, esta lista puede incluir todos los píxeles de referencia que se pueden usar para intra-predicción normal en HEVC.

En el ejemplo de la FIG. 4, las segundas paletas 192 incluyen cuatro entradas 208-214 que tienen el valor de índice de entrada 1, el valor de índice de entrada 2, el valor de índice de entrada 3 y el índice de entrada 4, respectivamente. Las entradas 208-214 relacionan los valores de índice con los valores de píxel, incluyendo el valor

de píxel A, el valor de píxel B, el valor de píxel C y el valor de píxel D, respectivamente. De acuerdo con uno o más aspectos de la presente divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden usar cualquiera de las técnicas descritas anteriormente para localizar la primera CU 180 con fines de predicción de paletas y copiar las entradas 1-3 de las primeras paletas 184 en las entradas 1-3 de las segundas paletas 192 para codificar la segunda CU 188. De esta manera, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar segundas paletas 192 basándose en primeras paletas 184. Además, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar datos para la entrada 4 a incluir con las segundas paletas 192. Dicha información puede incluir el número de entradas de paleta no predichas desde una paleta predictiva y los valores de píxel correspondientes a esas entradas de paleta.

En algunos ejemplos, de acuerdo con aspectos de la presente divulgación, uno o más elementos sintácticos pueden indicar si las paletas, tales como las segundas paletas 192, se predicen completamente a partir de una paleta predictiva (mostrada en la FIG. 4 como las primeras paletas 184, pero que puede estar compuesta de entradas de uno o más bloques) o si se predicen entradas particulares de las segundas paletas 192. Por ejemplo, un elemento sintáctico inicial puede indicar si se predicen todas las entradas. Si el elemento sintáctico inicial indica que no se predicen todas las entradas (por ejemplo, un indicador que tiene un valor de 0), uno o más elementos sintácticos adicionales pueden indicar qué entradas de las segundas paletas 192 se predicen a partir de la paleta predictiva.

De acuerdo con algunos aspectos de la presente divulgación, se puede deducir cierta información asociada con la predicción de paletas a partir de una o más características de los datos que se están codificando. Es decir, en lugar de que el codificador de vídeo 20 codifique elementos sintácticos (y el descodificador de vídeo 30 descodifique dichos elementos sintácticos), el codificador de vídeo 20 y el descodificador de vídeo 30 pueden realizar predicción de paletas basándose en una o más características de los datos que se están codificando.

En un ejemplo, con fines ilustrativos, se puede deducir el valor del indicador `_paleta_pred`, descrito anteriormente, a partir de uno o más de, como ejemplos, el tamaño de la CU que se está codificando, el tipo de trama, el espacio de color, el componente de color, el tamaño de la trama, la velocidad de tramas, el ID de capa en la codificación de vídeo escalable o el ID de vista en la codificación multivista. Es decir, con respecto al tamaño de la CU como un ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar que el indicador `_paleta_pred` descrito anteriormente es igual a uno para cualquier CU que exceda o sea inferior a un tamaño predeterminado. En este ejemplo, el indicador `_paleta_pred` no necesita señalizarse en el flujo de bits codificado.

Aunque se ha descrito anteriormente con respecto al indicador `_paleta_pred`, el codificador de vídeo 20 y/o el descodificador de vídeo 30 también pueden o de forma alternativa deducir otra información asociada con la predicción de paletas, tal como la CU candidata a partir de la que se usa la paleta para la predicción, o reglas para construir candidatos de predicción de paletas, basándose en una o más características de los datos que se están codificando.

De acuerdo con otros aspectos de la presente divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden construir una paleta sobre la marcha. Por ejemplo, cuando se codifica inicialmente la segunda CU 188, no hay entradas en las paletas 192. A medida que el codificador de vídeo 20 y el descodificador de vídeo 30 codifican valores nuevos para los píxeles de la segunda CU 188, cada valor nuevo se incluye en las paletas 192. Es decir, por ejemplo, el codificador de vídeo 20 añade valores de píxel a las paletas 192 a medida que se generan y se señalizan los valores de píxel para las posiciones en la CU 188. Como el codificador de vídeo 20 codifica píxeles relativamente más tarde en la CU, el codificador de vídeo 20 puede codificar píxeles que tienen los mismos valores que los ya incluidos en la paleta usando valores de índice en lugar de señalar los valores de píxel. De manera similar, cuando el descodificador de vídeo 30 recibe un nuevo valor de píxel (por ejemplo, señalado mediante el codificador de vídeo 20) para una posición en la segunda CU 188, el descodificador de vídeo 30 incluye el valor de píxel en las paletas 192. Cuando las posiciones de píxeles descodificadas relativamente más tarde en la segunda CU 188 tienen valores de píxel que se han añadido a las segundas paletas 192, el descodificador de vídeo 30 puede recibir información tal como, por ejemplo, valores de índice que identifican los valores de píxel correspondientes en las segundas paletas 192 para la reconstrucción de los valores de píxel de la segunda CU 188.

En algunos ejemplos, como se describe con mayor detalle a continuación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden mantener las paletas 184 y 192 en o por debajo de un tamaño de paleta máximo. De acuerdo con aspectos de la presente divulgación, si se alcanza un tamaño de paleta máximo, por ejemplo, cuando las segundas paletas 192 se construyen dinámicamente sobre la marcha, entonces el codificador de vídeo 20 y/o el descodificador de vídeo 30 realizan el mismo proceso para eliminar una entrada de las segundas paletas 192. Un ejemplo de proceso para eliminar entradas de paleta es una técnica de primero en entrar, primero en salir (FIFO) en la que el codificador de vídeo 20 y el descodificador de vídeo 30 eliminan la entrada más antigua de una paleta. En otro ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden eliminar la entrada de paleta usada con menos frecuencia de la paleta. En otro ejemplo adicional, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden ponderar tanto los procesos de FIFO como de frecuencia de uso para determinar qué entrada se debe eliminar. Es decir, la eliminación de una entrada puede basarse en la antigüedad de la entrada y con qué frecuencia se usa.

De acuerdo con algunos aspectos, si se elimina una entrada (valor de píxel) de una paleta y el valor de píxel ocurre de nuevo en una posición posterior en la CU que se está codificando, el codificador de vídeo 20 puede codificar el valor de píxel en lugar de incluir una entrada en la paleta y codificar un índice. De forma adicional o de forma alternativa, el codificador de vídeo 20 puede volver a introducir las entradas de paleta en la paleta después de haberse eliminado, por ejemplo, cuando el codificador de vídeo 20 y el descodificador de vídeo 30 exploran las posiciones en la CU.

En algunos ejemplos, las técnicas para obtener una paleta sobre la marcha se pueden combinar con una o más técnicas diferentes para determinar una paleta. En particular, como un ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar inicialmente las segundas paletas 192 (por ejemplo, usando la predicción de paletas para predecir las segundas paletas 192 a partir de las primeras paletas 184) y pueden actualizar las segundas paletas 192 cuando codifican píxeles de la segunda CU 188. Por ejemplo, tras transmitir la paleta inicial, el codificador de vídeo 20 puede añadir valores a la paleta inicial o cambiar valores en la paleta inicial cuando se exploran valores de píxel de localizaciones adicionales en la CU. Del mismo modo, tras recibir una paleta inicial, el descodificador de vídeo 30 puede añadir (es decir, incluir) valores a la paleta inicial o cambiar valores en la paleta inicial a medida que se exploran valores de píxel de localizaciones adicionales en la CU.

El codificador de vídeo 20 puede, en algunos ejemplos, señalar si la CU actual usa la transmisión de una paleta completa, o la generación de paletas sobre la marcha, o una combinación de transmisión de una paleta inicial con actualización de la paleta inicial mediante obtención sobre la marcha. En algunos ejemplos, la paleta inicial puede ser una paleta completa con el tamaño de paleta máximo, en cuyo caso se pueden cambiar los valores en la paleta inicial. En otros ejemplos, la paleta inicial puede ser menor que el tamaño de paleta máximo, en cuyo caso el codificador de vídeo 20 y el descodificador de vídeo 30 pueden añadir valores y/o cambiar valores de la paleta inicial.

De acuerdo con uno o más aspectos de la presente divulgación, el tamaño de las paletas, tales como las primeras paletas 184 y las segundas paletas 192, por ejemplo, en términos del número de valores de píxel que están incluidos en la paleta, puede fijarse o puede señalizarse usando uno o más elementos sintácticos en un flujo de bits codificado. Por ejemplo, de acuerdo con algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden usar códigos unarios o códigos unarios truncados (por ejemplo, códigos que truncan en un límite máximo del tamaño de paleta) para codificar el tamaño de paleta. De acuerdo con otros aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden usar códigos de Golomb exponenciales o de Rice-Golomb para codificar el tamaño de paleta.

De acuerdo con otros aspectos adicionales, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar datos que indican el tamaño de paleta después de cada entrada de la paleta. Con respecto a las segundas paletas 192 como un ejemplo, el codificador de vídeo 20 puede codificar un indicador de parada después de cada una de las entradas 208-214. En este ejemplo, un indicador de parada igual a uno puede especificar que la entrada que se está codificando actualmente es la entrada final de las segundas paletas 192, mientras que un indicador de parada igual a cero puede indicar que hay entradas adicionales en las segundas paletas 192. En consecuencia, el codificador de vídeo 20 puede codificar indicadores de parada que tienen un valor de cero después de cada una de las entradas 208-212 y un indicador de parada que tiene un valor de uno después de la entrada 214. En algunos casos, el indicador de parada puede no incluirse en el flujo de bits después de que la paleta construida alcanza un límite de tamaño de paleta máximo. Aunque los ejemplos anteriores divulgan técnicas para señalar explícitamente el tamaño de las paletas, en otros ejemplos, el tamaño de las paletas también se puede transmitir o deducir condicionalmente basándose en la denominada información lateral (por ejemplo, información característica tal como el tamaño de la CU que se está codificando, el tipo de trama, el espacio de color, el componente de color, el tamaño de trama, la velocidad de tramas, el ID de capa en la codificación de vídeo escalable o el ID de vista en la codificación multivista, como se ha observado anteriormente).

Las técnicas de la presente divulgación incluyen la codificación de datos sin pérdidas, o, de forma alternativa, con algunas pérdidas (codificación con pérdidas). Por ejemplo, con respecto a la codificación con pérdidas, el codificador de vídeo 20 puede codificar los píxeles de una CU sin hacer coincidir exactamente los valores de píxel de las paletas exactamente con los valores de píxel reales en la CU. Cuando las técnicas de la presente divulgación se aplican a la codificación con pérdidas, se pueden aplicar algunas restricciones a la paleta. Por ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden cuantizar paletas, tales como las primeras paletas 184 y las segundas paletas 192. Es decir, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden combinar o mezclar (es decir, cuantizar) entradas de una paleta cuando los valores de píxel de las entradas están dentro de un intervalo predeterminado entre sí. En otras palabras, si ya hay un valor de paleta que está dentro de un margen de error de un valor de paleta nuevo, el valor de paleta nuevo no se añade a la paleta. En otro ejemplo, una pluralidad de valores de píxel diferentes en un bloque puede asignarse a una única entrada de paleta, o, de forma equivalente, a un único valor de píxel de paleta.

El descodificador de vídeo 30 puede descodificar valores de píxel de la misma manera, independientemente de si una paleta particular no tiene pérdidas o tiene pérdidas. Como un ejemplo, el descodificador de vídeo 30 puede usar

un valor de índice transmitido mediante el codificador de vídeo 20 para una posición de píxel dada en un bloque codificado para seleccionar una entrada en la paleta para la posición de píxel, independientemente de si la paleta tiene pérdidas o no tiene pérdidas. En este ejemplo, el valor de píxel de la entrada de paleta se usa como el valor de píxel en el bloque codificado, tanto si coincide exactamente con el valor de píxel original como si no.

5 En un ejemplo de codificación con pérdidas, con fines ilustrativos, el codificador de vídeo 20 puede determinar un límite de error, denominado valor delta. Una entrada de valor de píxel candidato Cand_plt puede corresponder a un valor de píxel en una posición en un bloque a codificar, tal como una CU o PU. Durante la construcción de la paleta, el codificador de vídeo 20 determina la diferencia absoluta entre la entrada de valor de píxel candidato Cand_plt y
10 todas las entradas de valores de píxel existentes en la paleta. Si todas las diferencias absolutas entre la entrada de valor de píxel candidato Cand_plt y las entradas de valor de píxel existentes en la paleta son mayores que el valor delta, el codificador de vídeo 20 puede añadir el candidato de valor de píxel a la paleta como una entrada. Si una diferencia absoluta entre la entrada de valor de píxel Cand_plt y al menos una entrada de valor de píxel existente en la paleta es igual o menor que el valor delta, el codificador de vídeo 20 puede no añadir la entrada de valor de píxel
15 candidato Cand_plt a la paleta. Así pues, al codificar la entrada de valor de píxel Cand_plt, el codificador de vídeo 20 puede seleccionar la entrada con el valor de píxel que sea el más cercano a la entrada de valor de píxel Cand_plt, introduciendo de este modo ciertas pérdidas en el sistema. Cuando una paleta está formada por múltiples componentes (por ejemplo, tres componentes de color), la suma de la diferencia absoluta de los valores de componentes individuales se puede usar para comparar con el valor delta. De forma alternativa o de forma adicional,
20 la diferencia absoluta para cada valor de componente puede compararse con un segundo valor delta.

En algunos ejemplos, las entradas de valores de píxel existentes en la paleta observadas anteriormente pueden haberse añadido usando un proceso de comparación con delta similar. En otros ejemplos, los valores de píxel existentes en la paleta pueden haberse añadido usando otros procesos. Por ejemplo, una o más entradas de valores
25 de píxel iniciales se pueden añadir a una paleta (sin una comparación con delta) para iniciar el proceso de comparación con delta de construcción de la paleta. El procedimiento descrito anteriormente puede implementarse mediante el codificador de vídeo 20 y/o el decodificador de vídeo 30 para producir paletas de luma y/o croma.

Las técnicas descritas anteriormente con respecto a la construcción de la paleta también pueden usarse mediante el
30 codificador de vídeo 20 y el decodificador de vídeo 30 durante la codificación de píxeles. Por ejemplo, al codificar un valor de píxel, el codificador de vídeo 20 puede comparar el valor del píxel con los valores de píxel de entradas en la paleta. Si la diferencia absoluta del valor de píxel entre el valor del píxel y una de las entradas en la paleta es igual o menor que un valor delta, el codificador de vídeo 20 puede codificar el valor de píxel como la entrada de la paleta. Es decir, en este ejemplo, el codificador de vídeo 20 codifica el valor de píxel usando una de las entradas de
35 la paleta cuando el valor de píxel produce una diferencia absoluta suficientemente pequeña (por ejemplo, dentro de un intervalo predeterminado) frente a la entrada de paleta.

En algunos ejemplos, el codificador de vídeo 20 puede seleccionar la entrada de paleta que produce la diferencia absoluta del valor de píxel más pequeña (en comparación con el valor de píxel que se está codificando) para
40 codificar el valor de píxel. Como un ejemplo, el codificador de vídeo 20 puede codificar un índice para indicar una entrada de paleta que se usará para el valor de píxel, por ejemplo, la entrada del valor de píxel de paleta que se usará para reconstruir el valor de píxel codificado en el decodificador de vídeo 30. Si la diferencia absoluta del valor de píxel entre el valor de píxel y todas las entradas en la paleta es mayor que delta, el codificador puede no usar una de las entradas de paleta para codificar el valor de píxel, y en su lugar puede transmitir el valor de píxel del píxel
45 (posiblemente después de la cuantización) al decodificador de vídeo 30 (y posiblemente añadir el valor de píxel como una entrada a la paleta).

En otro ejemplo, el codificador de vídeo 20 puede seleccionar una entrada de una paleta para codificar un valor de píxel. El codificador de vídeo 20 puede usar la entrada seleccionada como un valor de píxel predictivo. Es decir, el
50 codificador de vídeo 20 puede determinar un valor residual que representa una diferencia entre el valor de píxel real y la entrada seleccionada y codificar el residuo. El codificador de vídeo 20 puede generar valores residuales para píxeles en un bloque que se predicen mediante entradas de una paleta, y puede generar un bloque de residuos que incluye valores de píxel residuales respectivos para el bloque de píxeles. El codificador de vídeo 20 puede aplicar posteriormente la transformación y la cuantización (como se ha observado anteriormente con respecto a la FIG. 2) al
55 bloque de residuos. De esta manera, el codificador de vídeo 20 puede generar coeficientes de transformación residuales cuantizados. En otro ejemplo, el residuo puede codificarse sin pérdidas (sin transformación y cuantización) o sin transformación.

El decodificador de vídeo 30 puede transformar de forma inversa y cuantizar inversamente los coeficientes de transformación para reproducir el bloque residual. El decodificador de vídeo 30 puede entonces reconstruir un valor
60 de píxel usando el valor de entrada de paleta predictiva y el valor residual para el valor de píxel. Por ejemplo, el decodificador de vídeo 30 puede combinar el valor residual con el valor de entrada de paleta para reconstruir el valor de píxel codificado.

65 En algunos ejemplos, el valor delta puede ser diferente para diferentes tamaños de CU, tamaños de imagen, espacios de color o componentes de color diferentes. El valor delta puede determinarse o determinarse

basándose en diversas condiciones de codificación. Por ejemplo, el codificador de vídeo 20 puede señalar el valor delta al descodificador de vídeo 30 usando sintaxis de alto nivel, tal como sintaxis en PPS, SPS, VPS y/o cabecera de fragmento. En otros ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden preconfigurarse para utilizar el mismo valor delta fijo. En otros ejemplos adicionales, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden obtener de forma adaptativa el valor delta basándose en información lateral (por ejemplo, tal como el tamaño de CU, el espacio de color, el componente de color o similares, como se ha observado anteriormente).

En algunos ejemplos, se puede incluir un modo de paleta de codificación con pérdidas como un modo de codificación HEVC. Por ejemplo, los modos de codificación pueden incluir un modo de intra-predicción, un modo de inter-predicción, un modo de paleta de codificación sin pérdidas, y un modo de paleta de codificación con pérdidas. En la codificación HEVC, como se ha observado anteriormente con respecto a las FIG. 2 y 3, se usa un parámetro de cuantización (QP) para controlar la distorsión permitida. El valor de delta para la codificación basada en paleta se puede calcular o de otro modo determinar como una función del QP. Por ejemplo, el valor delta descrito anteriormente puede ser $1 \ll (QP/6)$ o $1 \ll ((QP + d)/6)$ donde d es una constante, y " $\ll$ " representa el operador de desplazamiento a la izquierda a nivel de bit.

La generación de una paleta usando las técnicas de codificación con pérdidas descritas en la presente divulgación puede realizarse mediante el codificador de vídeo 20, el descodificador de vídeo 30 o ambos. Por ejemplo, el codificador de vídeo 20 puede generar entradas en una paleta para una CU usando las técnicas de comparación con delta descritas anteriormente y señalar información para la construcción de la paleta para su uso mediante el descodificador de vídeo 30. Es decir, el codificador de vídeo 20 puede configurarse para señalar información que indica valores de píxel para entradas en una paleta para una CU, y a continuación codificar valores de píxel usando los valores de píxel asociados con dichas entradas de paleta. El descodificador de vídeo 30 puede construir una paleta usando dicha información y, a continuación, puede usar las entradas para descodificar valores de píxel de un bloque codificado. En algunos ejemplos, el codificador de vídeo 20 puede señalar valores de índice que identifican entradas de paleta para una o más posiciones de píxel del bloque codificado, y el descodificador de vídeo 30 puede usar los valores de índice para recuperar las entradas de valores de píxel pertinentes de la paleta.

En otros ejemplos, el descodificador de vídeo 30 puede configurarse para construir una paleta aplicando las técnicas de comparación con delta descritas anteriormente. Por ejemplo, el descodificador de vídeo 30 puede recibir valores de píxel para posiciones dentro de un bloque codificado y puede determinar si las diferencias absolutas entre los valores de píxel y las entradas de valores de píxel existentes en la paleta son mayores que un valor delta. En dicho caso, el descodificador de vídeo 30 puede añadir los valores de píxel como entradas en la paleta, por ejemplo, para su uso posterior en la descodificación basada en paleta de valores de píxel para otras posiciones de píxeles del bloque usando los valores de índice correspondientes señalizados por el codificador 20 de vídeo. En este caso, el codificador de vídeo 20 y el descodificador de vídeo 30 aplican procesos iguales o similares para generar la paleta. En caso contrario, el descodificador de vídeo 30 puede no añadir los valores de píxel a la paleta.

En un ejemplo con fines ilustrativos, el descodificador de vídeo 30 puede recibir valores de índice o valores de píxel para varias posiciones de píxel en un bloque. Si se recibe un valor de índice para una posición de píxel, por ejemplo, el descodificador de vídeo 30 puede usar el valor de índice para identificar una entrada en la paleta y usar el valor de píxel de la entrada de paleta para la posición de píxel. Si se recibe un valor de píxel para la posición de píxel, el descodificador de vídeo 30 puede usar el valor de píxel recibido para la posición de píxel, y también puede aplicar la comparación con delta para determinar si el valor de píxel debe añadirse a la paleta y usarse posteriormente para la codificación de paleta.

En el lado del codificador, si un valor de píxel para una posición en un bloque produce una diferencia absoluta entre el valor de píxel y una entrada de valor de píxel existente en la paleta que es menor o igual que el valor delta, el codificador de vídeo 20 puede enviar un valor de índice para identificar la entrada en la paleta para su uso en la reconstrucción del valor de píxel para esa posición. Si un valor de píxel para una posición en un bloque produce valores de la diferencia absoluta entre el valor de píxel y las entradas de valores de píxel existentes en la paleta que son todos mayores que el valor delta, el codificador de vídeo 20 puede enviar el valor de píxel y puede añadir el valor de píxel como una entrada nueva en la paleta. Para construir la paleta, el descodificador de vídeo 30 puede usar valores delta señalizados mediante el codificador, basarse en un valor delta fijo o conocido, o deducir u obtener un valor delta, por ejemplo, como se ha descrito anteriormente.

Como se ha observado anteriormente, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden usar modos de codificación que incluyen un modo de intra-predicción, un modo de inter-predicción, un modo de paleta de codificación sin pérdidas, y un modo de paleta de codificación con pérdidas cuando codifican datos de vídeo. De acuerdo con algunos aspectos de la presente divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar uno o más elementos sintácticos que indican si la codificación basada en paleta está habilitada. Por ejemplo, en cada CU, el codificador de vídeo 20 puede codificar un elemento sintáctico, tal como un indicador indicador_Modo_PLT. El indicador_Modo_PLT u otro elemento sintáctico puede indicar si se debe usar un modo de codificación basado en paleta para una CU dada (o una PU en otros ejemplos). Por ejemplo, este indicador puede señalizarse en un flujo de bits de vídeo codificados en el nivel de CU, y a continuación recibirse mediante el

descodificador de vídeo 30 tras descodificar el flujo de bits de vídeo codificados.

En este ejemplo, un valor de este indicador_Modo_PLT igual a 1 puede especificar que la CU actual se codifica usando un modo de codificación basado en paleta. En este caso, el descodificador de vídeo 30 puede aplicar el modo de codificación basado en paleta para descodificar la CU. En algunos ejemplos, un elemento sintáctico puede indicar uno de una pluralidad de diferentes modos de codificación basados en paleta para la CU (por ejemplo, con pérdidas o sin pérdidas). Un valor de este indicador_Modo_PLT igual a 0 puede especificar que la CU actual se codifica usando un modo distinto del modo de paleta. Por ejemplo, puede usarse cualquiera de diversos modos de codificación inter-predictiva, intra-predictiva u otros. Cuando un valor de indicador_Modo_PLT es 0, el codificador de vídeo 20 también puede codificar datos adicionales para indicar el modo específico usado para codificar la CU respectiva (por ejemplo, un modo de codificación HEVC). El uso del indicador_Modo_PLT se describe a modo de ejemplo. Sin embargo, en otros ejemplos se pueden usar otros elementos sintácticos tales como códigos de varios bits para indicar si el modo de codificación basado en paleta se va a usar para una CU (o PU en otros ejemplos) o para indicar cuál de una pluralidad de modos se va a usar para la codificación.

En algunos ejemplos, el indicador descrito anteriormente u otros elementos sintácticos pueden transmitirse en un nivel superior al nivel de CU (o PU). Por ejemplo, el codificador de vídeo 20 puede señalar dicho indicador en un nivel de fragmento. En este caso, un valor igual a 1 indica que todas las CU del fragmento están codificadas usando el modo de paleta. En este ejemplo, ninguna información adicional del modo, por ejemplo, para el modo de paleta u otros modos, se señala en el nivel de CU. En otro ejemplo, el codificador de vídeo 20 puede señalar dicho indicador en un PPS, SPS o VPS.

De acuerdo con algunos aspectos de la presente divulgación, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar uno o más elementos sintácticos (por ejemplo, tal como el indicador descrito anteriormente) en uno de los niveles de fragmento, PPS, SPS o VPS especificando si el modo de paleta se habilita o deshabilita para el fragmento, imagen, secuencia o similar en particular, mientras que indicador_Modo_PLT indica si se usa el modo de codificación basado en paleta para cada CU. En este caso, si un indicador u otro elemento sintáctico enviado en el nivel de fragmento, PPS, SPS o VPS indica que el modo de codificación de paleta está deshabilitado, en algunos ejemplos, puede no ser necesario señalar el indicador_Modo_PLT para cada CU. De forma alternativa, si un indicador u otro elemento sintáctico enviado en el nivel de fragmento, PPS, SPS o VPS indica que el modo de codificación de paleta está habilitado, el indicador_Modo_PLT puede señalizarse de forma adicional para indicar si el modo de codificación basado en paleta debe usarse 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 de forma adicional o de forma alternativa para indicar la codificación basada en una paleta de una PU.

En algunos ejemplos, los elementos sintácticos descritos anteriormente pueden señalizarse condicionalmente en el flujo de bits. Por ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar o descodificar únicamente, respectivamente, los elementos sintácticos basándose en el tamaño de la CU, el tipo de trama, el espacio de color, el componente de color, el tamaño de trama, la velocidad de tramas, el ID de capa en la codificación de vídeo escalable o el ID de vista en la codificación multivista.

Aunque los ejemplos descritos anteriormente se refieren a señalización explícita, por ejemplo, con uno o más elementos sintácticos en un flujo de bits, en otros ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar implícitamente si un modo de codificación de paleta está activo y/o se usa para codificar un bloque particular. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden determinar si se usa una codificación basada en paleta para un bloque basándose en, por ejemplo, el tamaño de la CU, el tipo de trama, el espacio de color, el componente de color, el tamaño de trama, la velocidad de tramas, el ID de capa en la codificación de vídeo escalable o el ID de vista en la codificación multivista.

Aunque las técnicas de la FIG. 4 se han descrito anteriormente en el contexto de CU (HEVC), debe entenderse que las técnicas también pueden aplicarse a unidades de predicción (PU) o en otros procesos y/o normas de codificación de vídeo.

La FIG. 5 es un diagrama conceptual que ilustra ejemplos de determinación de índices para una paleta para un bloque de vídeo, coherente con las técnicas de la presente divulgación. Por ejemplo, la FIG. 5 incluye un mapa 240 de valores de índice (valores 1, 2 y 3) que relacionan posiciones respectivas de píxeles asociados con los valores de índice con una entrada de paletas 244. Las paletas 244 se pueden determinar de una manera similar a las primeras paletas 184 y las segundas paletas 192 descritas anteriormente con respecto a la FIG. 4.

Nuevamente, las técnicas de la FIG. 5 se describen en el contexto del codificador de vídeo 20 (FIG. 1 y FIG. 2) y el descodificador de vídeo 30 (FIG. 1 y FIG. 3) y con respecto a la norma de codificación de vídeo HEVC con fines explicativos. Sin embargo, debe entenderse que las técnicas de la presente divulgación no están limitadas de esta manera, y pueden aplicarse mediante otros procesadores y/o dispositivos de codificación de vídeo en otros procesos y/o normas de codificación de vídeo.

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

En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para codificar un mapa adicional que indica qué posiciones de píxel están asociadas con valores de índice. Por ejemplo, se supone que la entrada (i, j) en el mapa corresponde a la posición (i, j) de una CU. El codificador de vídeo 20 puede codificar uno o más elementos sintácticos para cada entrada del mapa (es decir, cada posición de píxel) indicando si la entrada tiene un valor de índice asociado. Por ejemplo, el codificador de vídeo 20 puede codificar un indicador que tiene un valor de uno para indicar que el valor de píxel en la localización (i, j) en la CU es uno de los valores en las paletas 244. El codificador de vídeo 20 puede, en dicho ejemplo, codificar también un índice de paleta (mostrado en el ejemplo de la FIG. 5 como los valores 1-3) para indicar ese valor de píxel en la paleta y para permitir que el descodificador de vídeo reconstruya el valor de píxel. En los casos en los que las paletas 244 incluyen una única entrada y un valor de píxel asociado, el codificador de vídeo 20 puede omitir la señalización del valor de índice. El codificador de vídeo 20 puede codificar el indicador para tener un valor de cero para indicar que el valor de píxel en la localización (i, j) en la CU no es uno de los valores en las paletas 244. En este ejemplo, el codificador de vídeo 20 también puede codificar una indicación del valor de píxel para su uso mediante el descodificador de vídeo 30 en la reconstrucción del valor de píxel. En algunos casos, el valor de píxel puede codificarse de una manera con pérdidas.

El valor de un píxel en una posición de una CU puede proporcionar una indicación de valores de uno o más píxeles diferentes en otras posiciones de la CU. Por ejemplo, puede haber una probabilidad relativamente alta de que las posiciones de píxel contiguas de una CU tengan el mismo valor de píxel o puedan asignarse al mismo valor de índice (en el caso de codificación con pérdidas, en la que se puede asignar más de un valor de píxel a un único valor de índice).

En consecuencia, de acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos indicando un número de píxeles consecutivos o valores de índice en un orden de exploración dado que tienen el mismo valor de píxel o valor de índice. Como se ha indicado anteriormente, la cadena de valores de índice o píxel de valor similar se puede denominar en el presente documento una serie. En un ejemplo con fines ilustrativos, si dos píxeles o índices consecutivos en un orden de exploración dado tienen valores diferentes, la serie es igual a cero. Si dos píxeles o índices consecutivos en un orden de exploración dado tienen el mismo valor pero el tercer píxel o índice en el orden de exploración tiene un valor diferente, la serie es igual a uno. Para tres índices o píxeles consecutivos con el mismo valor, la serie es dos, y así sucesivamente. El descodificador de vídeo 30 puede obtener los elementos sintácticos que indican una serie a partir de un flujo de bits codificado y usar los datos para determinar el número de localizaciones consecutivas que tienen el mismo valor de píxel o índice.

En algunos ejemplos, todas las localizaciones de píxel en la CU actual que tienen valores de píxel que están en la paleta para la CU actual se codifican con un índice de paleta seguido de una "serie" del valor de píxel en localizaciones de píxel consecutivas. En el caso de que solo haya una entrada en la paleta, la transmisión del índice de paleta o la "serie" se puede omitir para la CU actual. En el caso en que el valor de píxel en una de las localizaciones de píxel en la CU actual no tenga una coincidencia exacta con un valor de píxel en la paleta, el codificador de vídeo 20 puede seleccionar una de las entradas de paleta que tiene el valor de píxel más próximo y puede calcular un error de predicción o valor residual entre el valor de píxel original y el valor de píxel de predicción incluido en la paleta. El codificador de vídeo 20 codifica y transmite el valor residual para la localización de píxel al descodificador de vídeo. El descodificador de vídeo 30 puede entonces obtener un valor de píxel en la localización de píxel basándose en el índice de paleta recibido correspondiente, y el valor de píxel obtenido y el valor residual se usan entonces para predecir el valor de píxel original en la localización de píxel en la CU actual. En un ejemplo, el valor residual se codifica usando un procedimiento HEVC especificado por el Borrador 10 de HEVC, tal como aplicar un RQT para transformar el valor residual, cuantizar los coeficientes de transformación y codificar por entropía los coeficientes de transformación cuantizados. En algunos casos, el ejemplo anterior puede denominarse codificación con pérdidas.

En un ejemplo con fines ilustrativos, se considera la línea 248 del mapa 240. Suponiendo una dirección de exploración horizontal, de izquierda a derecha, la línea 248 incluye cinco valores de índice de "2" y tres valores de índice de "3". De acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 puede codificar un valor de índice de 2 para la primera posición de la línea 248 en la dirección de exploración. Además, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos que indican la serie de valores consecutivos en la dirección de exploración que tienen el mismo valor de índice que el valor de índice señalado. En el ejemplo de la línea 248, el codificador de vídeo 20 puede indicar una serie de 4, indicando con ello que los valores de índice de las siguientes cuatro posiciones en la dirección de exploración comparten el mismo valor de índice que el valor de índice señalado. El codificador de vídeo 20 puede realizar el mismo proceso para el siguiente valor de índice diferente en la línea 248. Es decir, el codificador de vídeo 20 puede codificar un valor de índice de 3 y uno o más elementos sintácticos que indican una serie de dos. El descodificador de vídeo 30 puede obtener los elementos sintácticos que

indican el valor de índice y el número de índices consecutivos en la dirección de exploración que tiene el mismo valor de índice (la serie).

Como se ha indicado anteriormente, los índices de un mapa se exploran en un orden particular. De acuerdo con aspectos de la presente divulgación, la dirección de exploración puede ser vertical, horizontal o diagonal (por ejemplo, 45 grados o 135 grados en diagonal en bloque). En algunos ejemplos, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos para cada bloque indicando una dirección de exploración para explorar los índices de los bloques. De forma adicional o de forma alternativa, la dirección de exploración puede señalizarse o deducirse basándose en la denominada información lateral tal como, por ejemplo, el tamaño del bloque, el espacio de color, y/o el componente de color. El codificador de vídeo 20 puede especificar exploraciones para cada componente de color de un bloque. De forma alternativa, una exploración especificada puede aplicarse a todos los componentes de color de un bloque.

Por ejemplo, con respecto a una exploración basada en columnas, se considera la columna 252 del mapa 240. Suponiendo una dirección de exploración vertical, de arriba a abajo, la columna 252 incluye un valor de índice de "1", cinco valores de índice de "2" y dos valores de índice de "3". De acuerdo con los aspectos de la presente divulgación, el codificador de vídeo 20 puede codificar un valor de índice de 1 para la primera posición de la línea 252 en la dirección de exploración (en la parte superior relativa de la columna 252). Además, el codificador de vídeo 20 puede señalizar una serie de cero, indicando con ello que el valor de índice de la siguiente posición en la dirección de exploración es diferente. El codificador de vídeo 20 puede entonces codificar un valor de índice de 2 para la siguiente posición en la dirección de exploración y uno o más elementos sintácticos que indican una serie de cuatro, es decir, que los valores de índice de las siguientes cuatro posiciones en la dirección de exploración comparten el mismo índice que el valor del índice señalizado. El codificador de vídeo 20 puede entonces codificar un valor de índice de 3 para el siguiente valor de índice diferente en la dirección de exploración y uno o más elementos sintácticos que indican una serie de uno. El descodificador de vídeo 30 puede obtener los elementos sintácticos que indican el valor de índice y el número de índices consecutivos en la dirección de exploración que tienen el mismo valor de índice (la serie).

De acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden realizar de forma adicional o de forma alternativa copiado de líneas para una o más entradas del mapa 240. El copiado de líneas puede depender, en algunos ejemplos, de la dirección de exploración. Por ejemplo, el codificador de vídeo 20 puede indicar que un valor de píxel o índice para una entrada particular en un mapa es igual a un valor de píxel o índice en una línea por encima de la entrada particular (para una exploración horizontal) o la columna a la izquierda de la entrada particular (para una exploración vertical). El codificador de vídeo 20 también puede indicar, como una serie, el número de valores de píxel o índice en el orden de exploración que son iguales a la entrada en la línea por encima o la columna a la izquierda de la entrada particular. En este ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden copiar valores de píxel o índice desde la línea contigua especificada y desde el número especificado de entradas para la línea del mapa que se está codificando actualmente.

En un ejemplo con fines ilustrativos, se consideran las columnas 256 y 260 del mapa 240. Suponiendo una dirección de exploración vertical, de arriba a abajo, la columna 256 incluye tres valores de índice de "1", tres valores de índice de "2" y dos valores de índice de "3". La columna 260 incluye los mismos valores de índice que tienen el mismo orden en la dirección de exploración. De acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos para la columna 260 indicando que toda la columna 260 se copia desde la columna 256. El uno o más elementos sintácticos pueden estar asociados con una primera entrada de la columna 260 en la parte superior relativa del mapa 240. El descodificador de vídeo 30 puede obtener los elementos sintácticos que indican el copiado de líneas y copiar los valores de índice de la columna 256 para la columna 260 cuando descodifica la columna 260.

De acuerdo con aspectos de la presente divulgación, las técnicas para codificar las denominadas series de entradas pueden usarse conjuntamente con las técnicas para copiado de líneas descritas anteriormente. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos (por ejemplo, un indicador) que indican si el valor de una entrada en un mapa se obtiene a partir de una paleta o el valor de una entrada en el mapa se obtiene a partir de una línea codificada previamente en el mapa 240. El codificador de vídeo 20 también puede codificar uno o más elementos sintácticos que indican un valor de índice de una paleta o la localización de la entrada en la línea (la fila o columna). El codificador de vídeo 20 también puede codificar uno o más elementos sintácticos que indican un número de entradas consecutivas que comparten el mismo valor. El descodificador de vídeo 30 puede obtener dicha información a partir de un flujo de bits codificados y puede usar la información para reconstruir los valores de mapa y de píxel para un bloque.

En un ejemplo con fines ilustrativos, se consideran las filas 264 y 268 del mapa 240. Suponiendo una dirección de exploración horizontal, de izquierda a derecha, la fila 264 incluye cinco valores de índice de "1" y tres valores de índice de "3". La fila 268 incluye tres valores de índice de "1", dos valores de índice de "2" y tres valores de índice de "3". En este ejemplo, el codificador de vídeo 20 puede identificar entradas particulares de la fila 264 seguidas por una serie cuando codifica datos para la fila 268. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más

elementos sintácticos que indican que la primera posición de la fila 268 (la posición del extremo izquierdo de la fila 268) es la misma que la primera posición de la fila 264. El codificador de vídeo 20 también puede codificar uno o más elementos sintácticos que indican que la siguiente serie de dos entradas consecutivas en la dirección de exploración en la fila 268 es la misma que la primera posición de la fila 264.

5 En algunos ejemplos, el codificador de vídeo 20 también puede determinar si debe codificar el valor de píxel o índice actual con respecto a una posición en otra fila (o columna) o codificar el valor de píxel o índice actual usando un elemento sintáctico de la serie. Por ejemplo, después de codificar uno o más elementos sintácticos que indican la primera posición de la fila 264 y la serie de dos entradas (observada anteriormente), el codificador de vídeo 20
10 puede codificar, para la cuarta y quinta posiciones en la línea 268 (de izquierda a derecha), uno o más elementos sintácticos que indican un valor de 2 para la cuarta posición y uno o más elementos sintácticos que indican una serie de 1. Por tanto, el codificador de vídeo 20 codifica estas dos posiciones sin referencia a otra línea (o columna). El codificador de vídeo 20 puede codificar entonces la primera posición que tiene un valor de índice de 3 en la fila 268 con respecto a la fila superior 264 (por ejemplo, indicando un copiado desde la fila superior 264 y la serie de
15 posiciones consecutivas en el orden de exploración que tienen el mismo valor de índice). Por lo tanto, de acuerdo con aspectos de la presente divulgación, el codificador de vídeo 20 puede seleccionar entre codificar valores de píxel o índice de una línea (o columna) con respecto a otros valores de la línea (o columna), por ejemplo, usando una serie, codificando valores de píxel o índice de una línea (o columna) con respecto a valores de otra línea (o columna), o una combinación de los mismos. En algunos ejemplos, el codificador de vídeo 20 puede realizar una
20 optimización de velocidad/distorsión para realizar la selección.

El descodificador de vídeo 30 puede recibir los elementos sintácticos descritos anteriormente y puede reconstruir la fila 268. Por ejemplo, el descodificador de vídeo 30 puede obtener datos que indican una localización particular en una fila contigua desde la que copiar el valor de índice asociado para la posición del mapa 240 que se está
25 codificando actualmente. El descodificador de vídeo 30 también puede obtener datos que indican el número de posiciones consecutivas en el orden de exploración que tienen el mismo valor de índice.

En algunos casos, la línea desde la que se copian las entradas puede ser directamente adyacente a la entrada de la línea que se está codificando actualmente (como se ilustra en los ejemplos de la FIG. 5). Sin embargo, en otros
30 ejemplos, un cierto número de líneas puede almacenarse en memoria intermedia mediante el codificador de vídeo 20 y/o el descodificador de vídeo 30, de tal manera que cualesquiera del cierto número de líneas del mapa pueden usarse como entradas predictivas para una línea del mapa que se está codificando actualmente. Por lo tanto, en algunos ejemplos, se puede señalar que el valor de píxel para una entrada es igual a un valor de píxel de una entrada en una fila inmediatamente por encima (o columna a la izquierda) de o dos o más filas por encima (o
35 columna a la izquierda) de la fila actual.

En un ejemplo con fines ilustrativos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden configurarse para almacenar las n filas de entradas anteriores antes de codificar una fila actual de entradas. En este ejemplo, el
40 codificador de vídeo 20 puede indicar la fila predictiva (la fila desde la que se copian las entradas) en un flujo de bits con un código unario truncado u otros códigos. En otro ejemplo, el codificador de vídeo 20 puede codificar (y el descodificador de vídeo 30 puede descodificar) un valor de desplazamiento entre la línea actual y la línea predictiva del mapa 240 usada como referencia para codificar la línea actual. Es decir, el codificador de vídeo 20 puede codificar una indicación de una línea (o columna) particular a partir de la que se copia un valor de índice. En algunos
45 ejemplos, el valor de desplazamiento puede ser un vector de desplazamiento. Es decir, se supone que $c[0]$, $c[1]$, ..., denotan los índices de la línea actual del mapa 240 y que $u[0]$, $u[1]$, $u[2]$, ..., denotan los índices de una línea predictiva del mapa 240, tal como una línea contigua superior. En este ejemplo, dado un vector de desplazamiento d , el valor de índice para $c[i]$ puede predecirse a partir de $u[i+d]$, o $u[i-d]$ para evitar que d tome valores negativos. El valor de d puede codificarse usando códigos unarios, unarios truncados, de Golomb exponenciales o de Golomb-Rice.

50 Como otro ejemplo, el codificador de vídeo 20 puede señalar una instrucción, tal como "copiar desde la mitad izquierda de la línea superior o" copiar desde la mitad derecha de la línea superior", indicando la línea contigua y el número o parte de las entradas de la línea contigua a copiar a la línea del mapa que se está codificando actualmente. Como ejemplo adicional, el mapa de valores de índice puede reordenarse antes de la codificación. Por
55 ejemplo, el mapa de valores de índice se puede girar 90, 180 o 270 grados, o girarse de arriba a abajo o de izquierda a derecha para mejorar la eficiencia de codificación.

En otros ejemplos, el codificador de vídeo 20 no puede transmitir series de valores de índice de valor similar del mapa 240 al descodificador de vídeo 30. En este caso, el codificador de vídeo 20 y/o el descodificador de vídeo 30
60 pueden obtener implícitamente los valores de las series. En un ejemplo, el valor de una serie puede ser un valor constante, por ejemplo, 4, 8, 16, o similar. En otro ejemplo, el valor de una serie puede depender de información lateral para el bloque actual de datos de vídeo que se está codificando tal como, por ejemplo, el tamaño del bloque, el parámetro de cuantización (QP), el tipo de trama, el componente de color, el formato de color (por ejemplo, 4:4:4, 4:2:2 o 4:2:0), el espacio de color (por ejemplo, YUV o RGB), la dirección de exploración y/u otros tipos de
65 información característica para el bloque actual. En el caso en que el valor de una serie depende del tamaño del bloque, la serie puede ser igual a la anchura del bloque actual, la altura del bloque actual, la mitad de la anchura (o

la mitad de la altura) del bloque actual, una fracción de la anchura y/o la altura del bloque actual, o un múltiplo de la anchura y/o la altura del bloque actual. En otro ejemplo, el codificador de vídeo 20 puede señalar el valor de una serie a descodificador de vídeo 30 usando sintaxis de alto nivel, tal como sintaxis en un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de serie (SPS), un conjunto de parámetros de vídeo (VPS) y/o una cabecera de fragmento.

De forma adicional o de forma alternativa, el codificador de vídeo 20 puede no necesitar siquiera transmitir el mapa 240 al descodificador de vídeo 30. En su lugar, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden obtener implícitamente una posición o localización de inicio de cada serie de valores de índice incluidos en el mapa 240. En un ejemplo, la norma de codificación de vídeo aplicada mediante el codificador de vídeo 20 y/o el descodificador de vídeo 30 puede determinar que una serie solo puede comenzar en ciertas localizaciones. Por ejemplo, la serie puede comenzar únicamente al principio de cada fila, o al principio de cada N filas de un bloque actual que se está codificando. La localización de inicio puede ser diferente para diferentes direcciones de exploración. Por ejemplo, si se usa la exploración vertical, la serie puede comenzar únicamente al principio de una columna o al principio de cada N columnas del bloque actual.

En otro ejemplo, la localización de inicio puede obtenerse dependiendo de la información lateral del bloque actual tal como, por ejemplo, el tamaño del bloque, el QP, el tipo de trama, el componente de color, el formato de color (por ejemplo, 4:4:4, 4:2:2 o 4:2:0), el espacio de color (por ejemplo, YUV o RGB), la dirección de exploración y/u otros tipos de información característica para el bloque actual. En el caso en el que la localización de inicio de una serie depende del tamaño del bloque, la localización de inicio puede ser el punto medio de cada fila y/o cada columna, o una fracción (por ejemplo, $1/n$, $2/n$, ... $(n-1)/n$) de cada fila y/o columna. En otro ejemplo, el codificador de vídeo 20 puede señalar la posición de inicio al descodificador de vídeo 30 usando sintaxis de alto nivel, tal como sintaxis en un PPS, un SPS, un VPS y/o una cabecera de fragmento.

En algunos ejemplos, la obtención implícita de la posición de inicio y la obtención implícita de la serie, descritas cada una de ellas anteriormente, pueden combinarse. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar que una serie de valores de índice de valor similar del mapa es igual a una distancia entre dos posiciones de inicio contiguas. En el caso en el que la posición de inicio es el comienzo (es decir, la primera posición) de cada fila de un bloque actual, entonces el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar que la longitud de la serie es igual a la longitud de una fila completa del bloque actual.

La FIG. 6 es un diagrama conceptual que ilustra ejemplos de la determinación de un borde geométrico 270, 272 o 274 de un bloque de vídeo usando una serie de índices de paleta para el componente de luma, submuestreados de forma adaptativa para los componentes de croma, de forma coherente con las técnicas de la presente divulgación. En la FIG. 6, las muestras de luma se ilustran como círculos sin relleno, y las muestras de croma se ilustran como una de las muestras de luma superpuestas con un símbolo x. La FIG. 6 ilustra ejemplos de diferentes valores de la serie para componentes de luma y croma basándose en una localización del borde geométrico 270, 272 o 274 del bloque de vídeo.

En algunos casos, se genera y comparte una paleta para múltiples componentes de color en el bloque actual y, en otros casos, se generan paletas independientes para uno o más de los componentes de color. En un caso, se puede generar una paleta para el componente de luma y se puede generar otra paleta para ambos componentes de croma. En cualquier caso, la información geométrica puede compartirse entre los componentes de color. Normalmente existe una alta correlación entre las localizaciones de los bordes de bloques colocados en diferentes componentes de color porque los componentes de croma pueden haberse submuestreado a partir de los componentes de luma de una manera predefinida, tal como el muestreo 4:2:2 o 4:2:0.

Por ejemplo, en la codificación basada en paletas, puede usarse la codificación de series para indicar información de geometría para el bloque actual porque un borde del bloque actual romperá la serie. En el caso del formato de croma 4:4:4, la serie se puede generar una vez y usarse para todos los componentes de color. La serie puede generarse basándose en uno de los componentes de color, o la serie puede generarse usando más de uno de los componentes de color. En el caso del formato de croma 4:2:2, la serie usada para el componente de luma puede submuestrearse horizontalmente por un factor de dos para su aplicación a los componentes de croma. En el caso del formato de croma 4:2:0, la serie usada para el componente de luma puede submuestrearse horizontal y verticalmente por un factor de dos para su aplicación a los componentes de croma.

En algunos casos, el procedimiento de submuestreo de la serie puede ser adaptable a un procedimiento de submuestreo de croma. En este caso, el valor de la serie submuestreada para los componentes de croma se puede calcular de manera diferente de acuerdo con la localización del borde, por ejemplo, el borde 270, 272 o 274, del bloque de vídeo como se muestra en la FIG. 6. En un primer ejemplo, la FIG. 6 ilustra un borde geométrico 270 entre dos bloques de vídeo contiguos que está colocado de tal manera que una serie para el componente de luma tiene un valor "1" en el bloque izquierdo y un valor de "3" en el bloque derecho. En este caso, la serie submuestreada para los componentes de croma tiene un valor de "1" tanto en el bloque izquierdo como en el bloque derecho. En un segundo ejemplo, la FIG. 6 ilustra un borde geométrico 272 entre dos bloques de vídeo contiguos que está colocado de tal manera que una serie para el componente de luma tiene un valor "2" tanto en el bloque izquierdo como en el

bloque derecho. En este caso, la serie submuestreada para los componentes de croma tiene un valor de "1" tanto en el bloque izquierdo como en el bloque derecho. En un tercer ejemplo, la FIG. 6 ilustra un borde geométrico 274 entre dos bloques de vídeo contiguos que está colocado de tal manera que una serie para el componente de luma tiene un valor "3" en el bloque izquierdo y un valor de "1" en el bloque derecho. En este caso, la serie submuestreada para los componentes de croma tiene un valor de "2" en el bloque izquierdo y un valor de "0" en el bloque derecho.

Además de la información geométrica, también puede ser posible tener una paleta única para el valor de píxel de todos los componentes de color. Por ejemplo, para cada localización de píxel en el bloque actual, los valores de píxel en tres componentes de color (por ejemplo, luma Y y ambos componentes de croma U y V) pueden formar un vector (es decir, un vector de color). A continuación, se puede formar una paleta seleccionando un cierto número de vectores para representar el bloque actual. Puede ser posible tener una paleta de valores de píxel para el componente de luma y otra paleta de valores de píxel para los componentes de croma. En algunos casos, también puede ser posible combinar los dos procedimientos de compartición de información geométrica y tener una única paleta de valores de píxel usando un vector de color.

En algunos ejemplos, el copiado de líneas descrito con más detalle en otra parte de la presente divulgación también puede funcionar con una única paleta. En este caso, el índice de paleta para cada localización de píxel se señala como igual al índice de paleta de la fila superior, si la exploración es horizontal, o la columna de la izquierda, si la exploración es vertical, y a continuación la serie asociada de índices de paleta también se copia desde la fila o columna anterior. Con una paleta compartida, una entrada de paleta puede ser una tripleta de (Y, U, V), de modo que los valores Y, U, V posteriores pueden reconstruirse a partir del índice de paleta. Los valores reconstruidos pueden servir como los valores de píxel descodificados o pueden servir como valores de predicción que se combinan con valores residuales para obtener los valores de píxel descodificados. En el formato croma 4:2:2 y el formato croma 4:2:0, los componentes de croma se han submuestreado en comparación con los componentes de luma. En el ejemplo de submuestreo 2:1, las posiciones de luma pueden estar en 0, 1, 2, ..., y las posiciones de croma pueden estar en 1, 3, 5, ... o pueden estar en 0, 2, 4, Para posiciones en las que no existen componentes de croma, los componentes U y V en la entrada de paleta pueden descartarse.

La FIG. 7 es un diagrama de flujo que ilustra un proceso de ejemplo para codificar datos de vídeo residuales de predicción usando un modo de codificación basado en paletas, coherente con las técnicas de la presente divulgación. El proceso de ejemplo ilustrado en la FIG. 7 se describe en el presente documento con respecto a la unidad de codificación basada en paletas 122 del codificador de vídeo 20 de la FIG. 2. En otros ejemplos, uno o más componentes diferentes o adicionales pueden realizar el proceso de ejemplo ilustrado en la FIG. 7.

El codificador de vídeo 20 recibe datos de vídeo de un bloque de vídeo actual que se va a codificar usando codificación de vídeo basada en paletas de bloques de vídeo predichos, y envía los datos de vídeo a la unidad de codificación basada en paletas 122. La unidad de codificación basada en paletas 122 determina los valores residuales de predicción para el bloque actual basándose en valores de píxel del bloque actual y valores de píxel de referencia codificados previamente (280).

La unidad de codificación basada en paletas 122 puede calcular los valores residuales de predicción usando cualquier modo de predicción, por ejemplo, un modo de inter-predicción o un modo de intra-predicción de la norma HEVC. En un ejemplo, la unidad de codificación basada en paletas 122 puede usar la unidad de procesamiento de inter-predicción 120 para predecir valores de píxel del bloque actual usando valores de píxel codificados previamente en un bloque de referencia. En otro ejemplo, la unidad de codificación basada en paletas 122 puede usar la unidad de procesamiento de intra-predicción 126 para predecir valores de píxel del bloque actual usando valores de píxel codificados previamente en el bloque actual.

En algunos casos, la unidad 122 de codificación basada en paletas puede determinar los valores residuales de predicción para el bloque actual usando solo un subconjunto de procesos del modo de predicción. Por ejemplo, en el caso del modo de intra-predicción, los procesos de predicción DC, horizontal y/o vertical pueden estar habilitados, pero otros procesos del modo de intra-predicción pueden estar deshabilitados. Entre los procesos deshabilitados se puede incluir el filtrado en el modo de intra-predicción, por ejemplo, uno o más de MDIS, interpolación bilineal 1/32 pel, filtro de bordes o el filtro DC. Como un ejemplo adicional, en el caso del modo de inter-predicción, el proceso de promediado de píxeles, por ejemplo, una o más de la predicción ponderada, la bi-predicción o la interpolación sub-pel, puede deshabilitarse.

En un ejemplo, los valores residuales de predicción para el bloque actual pueden ser valores de píxel residuales para el bloque actual. En este ejemplo, la unidad de codificación basada en paletas 122 calcula los valores de píxel residuales para el bloque actual a partir de los valores de píxel del bloque actual y los valores de píxel de referencia codificados previamente. La unidad de codificación basada en paletas 122 procede entonces a codificar los valores de píxel residuales para el bloque actual usando la codificación de vídeo basada en paletas como se describe en las siguientes etapas.

En otro ejemplo, los valores residuales de predicción para el bloque actual pueden ser valores de coeficientes de transformación cuantizados residuales para el bloque actual. En este ejemplo, la unidad de codificación basada en

paletas 122 calcula los valores de píxel residuales para el bloque actual a partir de los valores de píxel del bloque actual y los valores de píxel de referencia codificados previamente, y envía los valores de píxel residuales a la unidad de procesamiento de transformación 104 y la unidad de cuantización 106 para que se transformen y cuanticen en valores de coeficientes de transformación cuantizados residuales para el bloque actual. La unidad de

5 codificación basada en paletas 122 procede entonces a codificar los valores de coeficientes de transformación cuantizados residuales para el bloque actual usando la codificación de vídeo basada en paletas como se describe en las siguientes etapas.

10 La unidad de codificación basada en paletas 122 genera una paleta para el bloque actual que incluye una o más entradas que indican los valores residuales de predicción para el bloque actual (282). La unidad de codificación basada en paletas 122 asigna uno o más de los valores residuales de predicción para el bloque actual a valores de índice que identifican entradas en la paleta usadas para representar los valores residuales de predicción en la paleta para el bloque actual (284). La unidad de codificación basada en paletas 122 codifica los valores de índice para una o más posiciones en el bloque actual (286). Los valores de índice codificados indican los valores residuales de

15 predicción incluidos en la paleta para el bloque actual que se usan para representar los valores residuales de predicción para el bloque actual. El codificador de vídeo 20 transmite entonces los valores de índice para la una o más posiciones en el bloque actual.

20 La FIG. 8 es un diagrama de flujo que ilustra un proceso de ejemplo para descodificar datos de vídeo residuales de predicción usando un modo de codificación basado en paletas, coherente con las técnicas de la presente divulgación. El proceso de ejemplo ilustrado en la FIG. 8 se describe en el presente documento con respecto a la unidad de descodificación basada en paletas 165 del descodificador de vídeo 30 de la FIG. 3. En otros ejemplos, uno o más componentes diferentes o adicionales pueden realizar el proceso de ejemplo ilustrado en la FIG. 8.

25 El descodificador de vídeo 30 recibe un flujo de bits que representa datos de vídeo codificados usando codificación basada en paletas, y envía los datos de vídeo descodificados por entropía a la unidad de descodificación basada en paletas 165. Basándose en uno o más elementos sintácticos incluidos en el flujo de bits descodificado, la unidad de descodificación basada en paletas 165 genera una paleta para un bloque actual de datos de vídeo que incluye una o más entradas que indican valores residuales de predicción para el bloque actual 290. La unidad de descodificación

30 basada en paletas 165 descodifica entonces valores de índice para una o más posiciones en el bloque actual (292). Los valores de índice descodificados indican los valores residuales de predicción incluidos en la paleta para el bloque actual que se usan para representar los valores residuales de predicción para el bloque actual.

35 La unidad de descodificación basada en paletas 165 determina uno o más de los valores residuales de predicción para el bloque actual basándose en los valores de índice que identifican entradas en la paleta que representan los valores residuales de predicción para el bloque actual (294). La unidad de descodificación basada en paletas 165 puede determinar los valores residuales de predicción usando cualquier modo de predicción, por ejemplo, un modo de inter-predicción o un modo de intra-predicción de la norma HEVC. En un ejemplo, la unidad de descodificación basada en paletas 165 puede usar la unidad de compensación de movimiento 164 para predecir valores de píxel del

40 bloque actual usando valores de píxel codificados previamente en un bloque de referencia. En otro ejemplo, la unidad de descodificación basada en paletas 165 puede usar la unidad de procesamiento de intra-predicción 166 para predecir valores de píxel del bloque actual usando valores de píxel codificados previamente en el bloque actual.

45 En algunos casos, la unidad de descodificación basada en paletas 165 puede determinar los valores residuales de predicción para el bloque actual usando solo un subconjunto de procesos del modo de predicción. Por ejemplo, en el caso del modo de intra-predicción, los procesos de predicción DC, horizontal y/o vertical pueden estar habilitados, pero otros procesos del modo de intra-predicción pueden estar deshabilitados. Los procesos deshabilitados pueden incluir el filtrado en el modo de intra-predicción, por ejemplo, uno o más de MDIS, interpolación bilineal 1/32 pel, filtro de bordes o filtro DC. Como un ejemplo adicional, en el caso del modo de inter-predicción, el proceso de promediado de píxeles, por ejemplo, una o más de la predicción ponderada, la bi-predicción o la interpolación sub-pel, puede estar deshabilitado.

50 El descodificador de vídeo 30 determina entonces los valores de píxel del bloque actual basándose en los valores residuales de predicción para el bloque actual y los valores de píxel de referencia codificados previamente (296). En un ejemplo, los valores residuales de predicción para el bloque actual pueden ser valores de píxel residuales para el bloque actual. En este caso, el descodificador de vídeo 30 reconstruye los valores de píxel del bloque actual usando los valores de píxel residuales y los valores de píxel de referencia codificados previamente. En otro ejemplo, los valores residuales de predicción para el bloque actual pueden ser valores de coeficientes de transformación cuantizados residuales para el bloque actual. En este caso, la unidad de descodificación basada en paletas 165

60 envía en primer lugar los valores de coeficientes de transformación cuantizados residuales a la unidad de cuantización inversa 154 y la unidad de procesamiento de transformación inversa 156 para que se cuanticen de forma inversa y se transformen de forma inversa en valores de píxel residuales para el bloque actual. El descodificador de vídeo 30 reconstruye entonces los valores de píxel del bloque actual usando los valores de píxel residuales y los valores de píxel de referencia codificados previamente.

65 La FIG. 9 es un diagrama de flujo que ilustra un proceso de ejemplo para generar una paleta para la codificación

basada en paletas, coherente con las técnicas de la presente divulgación. El proceso de ejemplo ilustrado en la FIG. 9 se describe en el presente documento con respecto a la unidad de descodificación basada en paletas 165 del descodificador de vídeo 30 de la FIG. 3. En otros ejemplos, el proceso también puede realizarse mediante la unidad de codificación basada en paletas 122 del codificador de vídeo 20 de la FIG. 2. El proceso de ejemplo para generar una paleta para la codificación basada en paletas se puede usar para generar una paleta que incluye entradas de paleta que indican valores de píxel. En otros ejemplos, puede usarse un proceso similar para generar una paleta que incluye entradas de paleta que indican valores residuales de predicción.

El descodificador de vídeo 30 recibe un flujo de bits que representa datos de vídeo codificados usando una codificación basada en paletas, y envía los datos de vídeo descodificados por entropía a la unidad de descodificación basada en paletas 165. La unidad de descodificación basada en paletas 165 genera una paleta predictiva que incluye entradas de paleta que indican valores de píxel (300). En algunos ejemplos, la unidad de descodificación basada en paletas 165 genera la paleta predictiva que incluye entradas de paleta de uno o más bloques codificados previamente de los datos de vídeo. Los bloques codificados previamente pueden incluir bloques contiguos de un bloque actual incluyendo bloques espacialmente contiguos y/o bloques contiguos en un orden de exploración particular de los bloques.

La unidad de descodificación basada en paletas 165 determina a continuación, a partir de los datos de vídeo descodificados por entropía, una o más de las entradas de paleta en la paleta predictiva que se copian a una paleta actual para el bloque actual (302). De manera más específica, la unidad de descodificación basada en paletas 165 puede descodificar uno o más elementos sintácticos que indican si cada una de las entradas de paleta en la paleta predictiva se copia en la paleta actual. En un ejemplo, el uno o más elementos sintácticos comprenden un vector binario que incluye un indicador para cada una de las entradas de paleta en la paleta predictiva que indica si una entrada de paleta respectiva se copia en la paleta actual. En otro ejemplo, el uno o más elementos sintácticos comprenden una versión comprimida sin pérdidas del vector binario, donde una versión sin comprimir del vector binario incluye un indicador para cada una de las entradas de paleta en la paleta predictiva que indica si una entrada de paleta respectiva se copia en la paleta actual.

La unidad de descodificación basada en paletas 165 también determina, a partir de los datos de vídeo descodificados por entropía, un cierto número de entradas de paleta nuevas no incluidas en la paleta predictiva que están incluidas en la paleta actual para el bloque actual (304). La unidad de descodificación basada en paletas 165 puede descodificar uno o más elementos sintácticos que indican el número de las entradas de paleta nuevas que se incluyen en la paleta actual. En algunos ejemplos, la unidad de descodificación basada en paletas descodifica los elementos sintácticos usando uno de códigos unarios, códigos unarios truncados, códigos de Golomb exponenciales, o códigos de Golomb-Rice. Después de determinar el número de entradas de paleta nuevas que se incluyen en la paleta actual, la unidad de descodificación basada en paletas 165 descodifica uno o más elementos sintácticos que indican un valor de píxel para cada una de las entradas de paleta nuevas.

Basándose en la información determinada a partir de los datos de vídeo descodificados por entropía, la unidad de descodificación basada en paletas 165 calcula un tamaño de la paleta actual igual a la suma del número de entradas de paleta copiadas y el número de las entradas de paleta nuevas (306). Después de determinar el tamaño de la paleta actual, la unidad de descodificación basada en paletas 165 genera la paleta actual que incluye las entradas de paleta copiadas y las entradas de paleta nuevas (308). En un ejemplo, la unidad de descodificación basada en paletas 165 genera la paleta actual concatenando las entradas de paleta copiadas y las entradas de paleta nuevas.

La unidad de descodificación basada en paletas 165 es entonces capaz de realizar la codificación basada en paletas del bloque actual usando la paleta actual. Por ejemplo, la unidad de descodificación basada en paletas 165 determina valores de índice para uno o más valores de píxel del bloque actual que identifican las entradas de paleta en la paleta actual usadas para representar los valores de píxel del bloque actual 310. En el caso en el que uno o más valores de píxel del bloque actual no tienen un valor de píxel correspondiente en la paleta actual, el codificador de vídeo 20 puede usar el concepto de píxel de escape para indicar cuáles de los valores de píxel no están incluidos en la paleta actual, y transmitir explícitamente estos valores de píxel. La unidad de descodificación basada en paletas 165 en el descodificador de vídeo 30 puede descodificar entonces uno o más elementos sintácticos que indican los valores de píxel para el uno o más valores de píxel que no tienen un valor de píxel correspondiente en la paleta actual.

En otro ejemplo, el codificador de vídeo 20 puede no usar el concepto de píxel de escape, pero en su lugar puede identificar valores de píxel incluidos en la paleta actual como valores de píxel de predicción para cada uno del uno o más valores de píxel de los bloques actuales, y transmitir valores residuales entre los valores de píxel del bloque actual y los valores de píxel de predicción en la paleta actual. La unidad de descodificación basada en paletas 165 en el descodificador de vídeo 30 puede entonces descodificar uno o más elementos sintácticos que indican los valores de índice que identifican los valores de píxel de predicción correspondientes incluidos en la paleta actual, y los valores residuales entre el uno o más valores de píxel del bloque actual y los valores de píxel de predicción identificados en la paleta actual.

La presente divulgación también describe varias técnicas alternativas para generar una paleta para la codificación

basada en paletas, que puede usarse para generar una paleta que tiene entradas que asocian valores de índice con los valores de píxel o valores residuales de predicción para un bloque actual. En un ejemplo, la unidad de descodificación basada en paletas 165 descodifica una indicación de un tamaño de la paleta para el bloque actual, descodifica un vector que tiene el mismo tamaño que la paleta para el bloque actual, donde cada entrada en el vector indica si una entrada de paleta asociada se transmite o copia desde una paleta predictiva y, para la una o más entradas de paleta copiadas desde la paleta predictiva, descodifica una indicación de una posición de la entrada en la paleta predictiva. En otro ejemplo, la unidad de descodificación basada en paletas 165 descodifica una indicación de un número de entradas en la paleta para el bloque actual, descodifica un indicador de un bit para cada una de las entradas de paleta que indica si la entrada de paleta se envía explícitamente o se obtiene a partir de un píxel reconstruido previamente y, para cada una de la una o más entradas de paleta obtenidas a partir del píxel reconstruido previamente, descodifica una indicación de una posición del píxel reconstruido que corresponde a la entrada de paleta respectiva. En ese ejemplo, la indicación de la posición del píxel reconstruido puede ser un vector de desplazamiento con respecto a la posición superior izquierda del bloque actual o puede ser un índice en una lista de píxeles reconstruidos que puede incluir todos los píxeles de referencia usados para la intra-predicción normal.

En otro ejemplo, la unidad de descodificación basada en paletas 165 comienza con una paleta predictiva para un bloque contiguo que tiene un tamaño dado, y descodifica un vector binario que tiene el mismo tamaño que la paleta predictiva, donde cada entrada en el vector indica si una entrada de paleta asociada se reutiliza a partir de la paleta predictiva. La unidad de descodificación basada en paletas 165 también descodifica una indicación del número de entradas nuevas a transmitir, y recibe las entradas nuevas del codificador de vídeo 20. La unidad de descodificación basada en paletas 165 combina entonces las entradas reutilizadas y las entradas nuevas para generar la paleta nueva para el bloque actual.

Debe reconocerse que dependiendo del ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en el presente documento pueden realizarse en una secuencia distinta, pueden añadirse, combinarse u omitirse por completo (por ejemplo, no todos los actos o eventos descritos son necesarios para llevar a la práctica las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse simultáneamente, por ejemplo, mediante el procesamiento de múltiples hilos, el procesamiento de interrupciones, o múltiples procesadores, en lugar de secuencialmente. Además, aunque ciertos aspectos de la presente divulgación se describen como realizados por un único módulo o unidad con fines de claridad, se debería entender que las técnicas de la presente divulgación pueden realizarse mediante una combinación de unidades o módulos asociados con un codificador de vídeo.

Se han descrito ciertos aspectos de la presente divulgación con respecto al desarrollo de la norma HEVC con fines ilustrativos. Sin embargo, las técnicas descritas en la presente divulgación pueden ser útiles para otros procesos de codificación de vídeo, incluyendo otros procesos de codificación de vídeo estándares o patentados aún no desarrollados.

Las técnicas descritas anteriormente pueden realizarse mediante el codificador de vídeo 20 (FIG. 1 y 2) y/o el descodificador de vídeo 30 (FIG. 1 y 3), donde ambos pueden denominarse en general un codificador de vídeo. Del mismo modo, la codificación de vídeo puede referirse a la codificación de vídeo o a la descodificación de vídeo, según sea aplicable.

Aunque se han descrito anteriormente combinaciones particulares de diversos aspectos de las técnicas, estas combinaciones se proporcionan simplemente para ilustrar ejemplos de las técnicas descritas en la presente divulgación. En consecuencia, las técnicas de la presente divulgación no deben limitarse a estas combinaciones de ejemplo y pueden abarcar cualquier combinación concebible de los diversos aspectos de las técnicas descritas en la presente divulgación.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones, como una o más instrucciones o código, pueden almacenarse en, o transmitirse mediante, un medio legible por ordenador, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde 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 y legibles por ordenador, que sean no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualesquiera medios disponibles a los que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para implementar las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo, y no de manera limitativa, 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 pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que

pueda accederse mediante un ordenador. Además, cualquier conexión puede denominarse adecuadamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco de láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente reproducen datos de manera magnética, mientras que otros discos reproducen los datos de manera óptica con láser. Las combinaciones de lo anterior también deberían incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ejecutarse mediante uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables por campo (FPGA), u otros circuitos lógicos integrados o discretos equivalentes. En consecuencia, el término "procesadores", 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 puede proporcionarse dentro de módulos de hardware y/o software específicos configurados para la codificación y la decodificación, o incorporarse en un codec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En la presente divulgación se describen varios componentes, módulos o unidades para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente la realización mediante diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, pueden combinarse diversas unidades en una unidad de hardware de codec, o proporcionarse mediante una colección de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, conjuntamente con el software y/o firmware adecuado.

Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:
 - la generación de una paleta predictiva que incluye entradas de paleta que indican valores de píxel;
 - la determinación de cuáles de las entradas de paleta en la paleta predictiva se deben copiar en una paleta actual para un bloque actual de los datos de vídeo,
 - la codificación de uno o más elementos sintácticos que indican un número de entradas de paleta nuevas que se deben incluir en la paleta actual para el bloque actual, en el que las entradas de paleta nuevas no están en la paleta predictiva;
 - la codificación de uno o más elementos sintácticos que indican un valor de píxel para cada una de las entradas de paleta nuevas a incluir en la paleta actual;
 - el cálculo de un tamaño de la paleta actual igual a la suma de un número de las entradas de paleta copiadas y el número de las entradas de paleta nuevas;
 - la generación de la paleta actual que tiene el tamaño calculado que incluye las entradas de paleta copiadas y las entradas de paleta nuevas; y
 - la determinación de valores de índice para uno o más valores de píxel del bloque actual que identifican las entradas de paleta en la paleta actual usadas para representar los valores de píxel del bloque actual, **caracterizado por que:**
 - la determinación de las entradas de paleta que se deben copiar comprende la codificación de uno o más elementos sintácticos que indican si cada una de las entradas de paleta en la paleta predictiva se deben copiar en la paleta actual.
2. El procedimiento de acuerdo con la reivindicación 1, en el que la generación de la paleta predictiva comprende la generación de la paleta predictiva incluyendo entradas de paleta de uno o más bloques codificados previamente de los datos de vídeo.
3. El procedimiento de acuerdo con la reivindicación 2, en el que uno o más bloques codificados previamente comprenden bloques contiguos del bloque actual que incluyen al menos uno de bloques contiguos espacialmente o bloques contiguos en un orden de exploración particular de los bloques.
4. El procedimiento de acuerdo con la reivindicación 1, en el que el uno o más elementos sintácticos que indican si cada una de las entradas de paleta en la paleta predictiva se deben copiar en la paleta actual comprenden un vector binario que incluye un indicador para cada una de las entradas de paleta en la paleta predictiva que indica si una entrada de paleta respectiva se debe copiar en la paleta actual.
5. El procedimiento de acuerdo con la reivindicación 1, en el que uno o más elementos sintácticos que indican si cada una de las entradas de paleta en la paleta predictiva se deben copiar en la paleta actual comprenden una versión comprimida sin pérdidas de un vector binario, en el que una versión sin comprimir del vector binario incluye un indicador para cada una de las entradas de paleta en la paleta predictiva que indica si una entrada de paleta respectiva se debe copiar en la paleta actual.
6. El procedimiento de acuerdo con la reivindicación 1, en el que el uno o más elementos sintácticos que indican el número de las entradas de paleta nuevas que deben incluirse en la paleta actual se codifican usando al menos uno de códigos unarios, códigos unarios truncados, códigos de Golomb exponenciales o códigos de Golomb-Rice.
7. El procedimiento de acuerdo con la reivindicación 1, en el que la generación de la paleta actual que incluye las entradas de paleta copiadas y las entradas de paleta nuevas comprende la concatenación de las entradas de paleta copiadas y las entradas de paleta nuevas.
8. El procedimiento de acuerdo con la reivindicación 1, en el que el bloque actual comprende uno de un primer bloque en un fragmento de los datos de vídeo o un bloque del extremo izquierdo del fragmento o una imagen de los datos de vídeo, comprendiendo además el procedimiento la deshabilitación del copiado de entradas de paleta de la paleta predictiva en la paleta actual para el bloque actual.
9. El procedimiento de acuerdo con la reivindicación 1, en el que la determinación de los valores de índice para uno o más valores de píxel del bloque actual comprende:
 - la codificación de uno o más elementos sintácticos que indican si cada uno de los valores de píxel del

bloque actual tiene un valor de píxel correspondiente en la paleta actual;

la codificación de uno o más elementos sintácticos que indican los valores de índice para el uno o más valores de píxel del bloque actual que tienen valores de píxel correspondientes en la paleta actual; y

la codificación de uno o más elementos sintácticos que indican los valores de píxel para el uno o más valores de píxel del bloque actual que no tienen un valor de píxel correspondiente en la paleta actual.

10. El procedimiento de acuerdo con la reivindicación 1, en el que la determinación de los valores de índice para el uno o más valores de píxel del bloque actual comprende:

la codificación de uno o más elementos sintácticos que indican los valores de índice para el uno o más valores de píxel del bloque actual, en el que los valores de índice identifican valores de píxel correspondientes en la paleta actual como valores de píxel de predicción; y

la codificación de uno o más elementos sintácticos que indican valores residuales entre el uno o más valores de píxel del bloque actual y los valores de píxel de predicción identificados en la paleta actual.

11. El procedimiento de acuerdo con la reivindicación 1, en el que el procedimiento de codificación de datos de vídeo comprende un procedimiento de descodificación de datos de vídeo, comprendiendo el procedimiento:

la descodificación de uno o más elementos sintácticos que indican si cada una de las entradas de paleta en la paleta predictiva se deben copiar en la paleta actual;

la descodificación de uno o más elementos sintácticos que indican el número de las entradas de paleta nuevas que se deben incluir en la paleta actual;

la descodificación de uno o más elementos sintácticos que indican el valor de píxel para cada una de las entradas de paleta nuevas;

la generación de la paleta actual incluyendo las entradas de paleta copiadas y las entradas de paleta nuevas; y

la descodificación de uno o más elementos sintácticos que indican los valores de índice para el uno o más valores de píxel del bloque actual que identifican las entradas de paleta en la paleta actual usadas para representar los valores de píxel del bloque actual.

12. El procedimiento de acuerdo con la reivindicación 1, en el que el procedimiento de codificación de datos de vídeo comprende un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:

la generación de la paleta actual incluyendo las entradas de paleta copiadas y las entradas de paleta nuevas;

la codificación de uno o más elementos sintácticos que indican si cada una de las entradas de paleta en la paleta predictiva se debe copiar en la paleta actual;

la codificación de uno o más elementos sintácticos que indican el número de las entradas de paleta nuevas que se deben incluir en la paleta actual;

la codificación de uno o más elementos sintácticos que indican el valor de píxel para cada una de las entradas de paleta nuevas; y

la codificación de uno o más elementos sintácticos que indican los valores de índice para el uno o más valores de píxel del bloque actual que identifican las entradas de paleta en la paleta actual usadas para representar los valores de píxel del bloque actual.

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

medios para generar una paleta predictiva que incluye entradas de paleta que indican valores de píxel;

medios para determinar cuáles de las entradas de paleta en la paleta predictiva se deben copiar a una paleta actual para un bloque actual de los datos de vídeo,

medios para codificar uno o más elementos sintácticos que indican un número de entradas de paleta nuevas que se deben incluir en la paleta actual para el bloque actual, en el que las entradas de paleta nuevas no están en la paleta predictiva;

- medios para codificar uno o más elementos sintácticos que indican un valor de píxel para cada una de las entradas de paleta nuevas a incluir en la paleta actual;
- 5 medios para calcular un tamaño de la paleta actual igual a la suma de un número de las entradas de paleta copiadas y el número de las entradas de paleta nuevas;
- medios para generar la paleta actual que tiene el tamaño calculado que incluye las entradas de paleta copiadas y las entradas de paleta nuevas; y
- 10 medios para determinar valores de índice para uno o más valores de píxel del bloque actual que identifican las entradas de paleta en la paleta actual usadas para representar los valores de píxel del bloque actual, **caracterizado por que:**
- 15 los medios para determinar las entradas de paleta que se deben copiar comprenden medios para codificar uno o más elementos sintácticos que indican si cada una de las entradas de paleta en la paleta predictiva se debe copiar en la paleta actual.
14. Un medio legible por ordenador no transitorio que almacena instrucciones en el mismo que, al ejecutarse, hacen que uno o más procesadores realicen el procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 12.
- 20

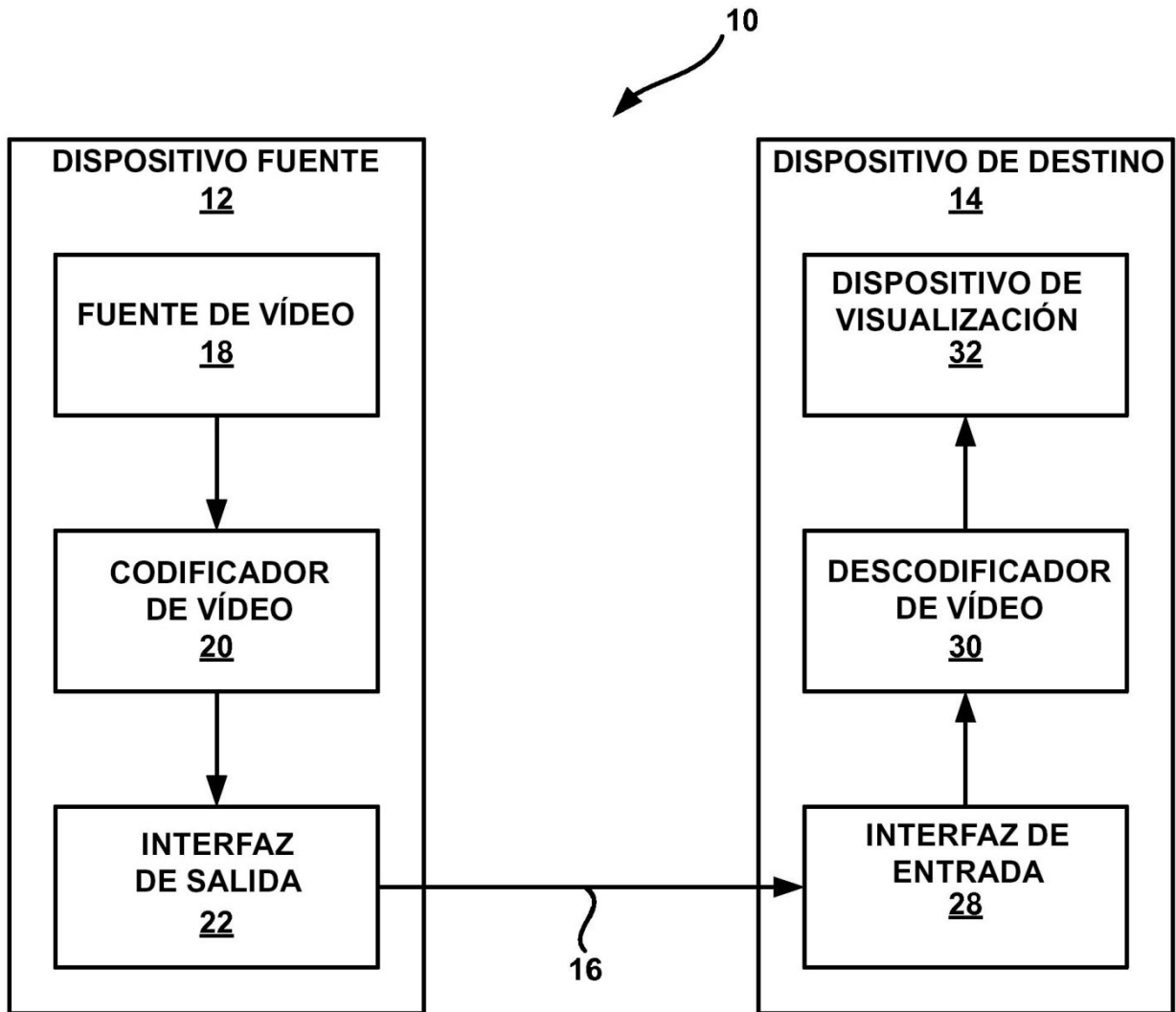


FIG. 1

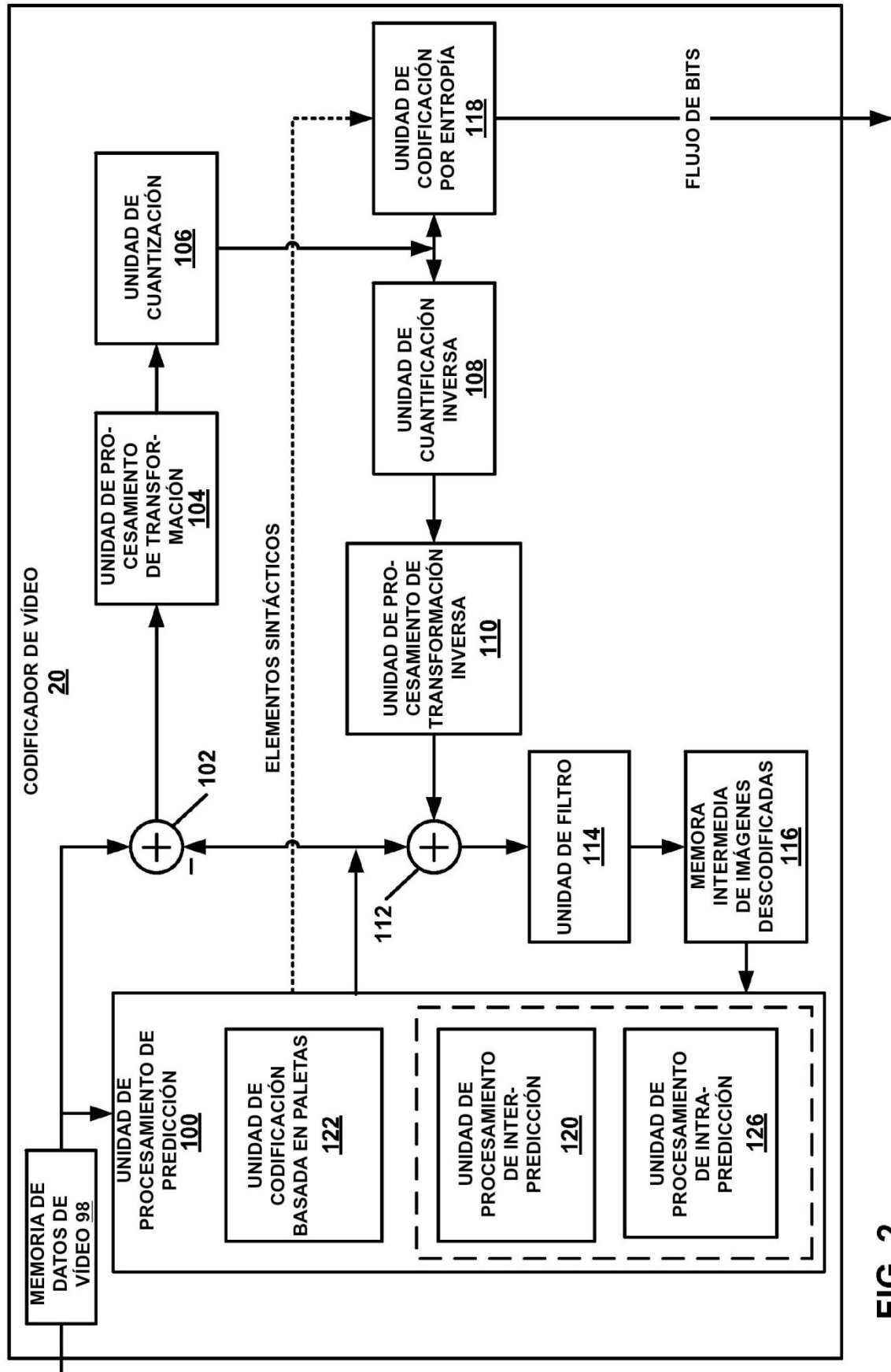
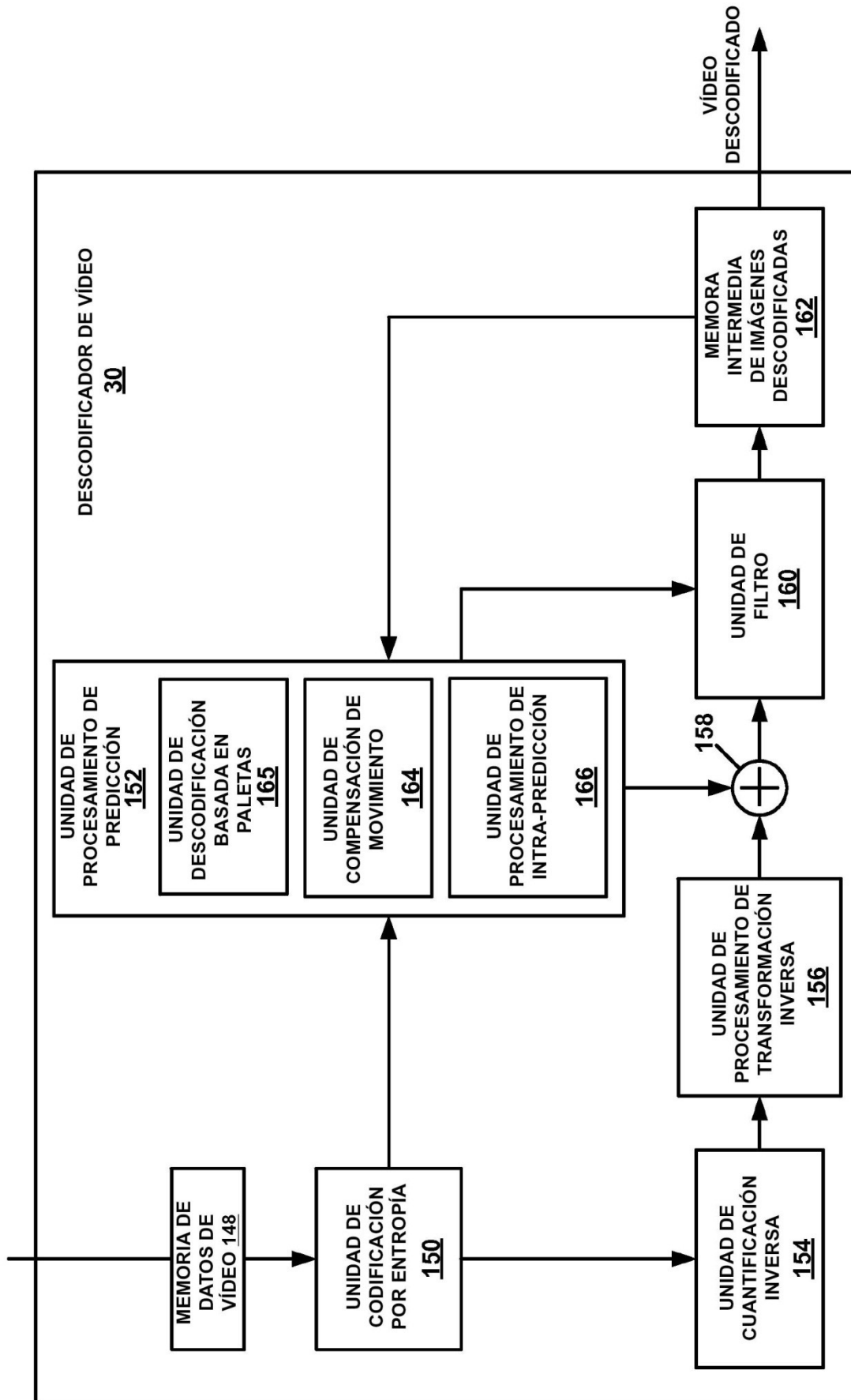


FIG. 2



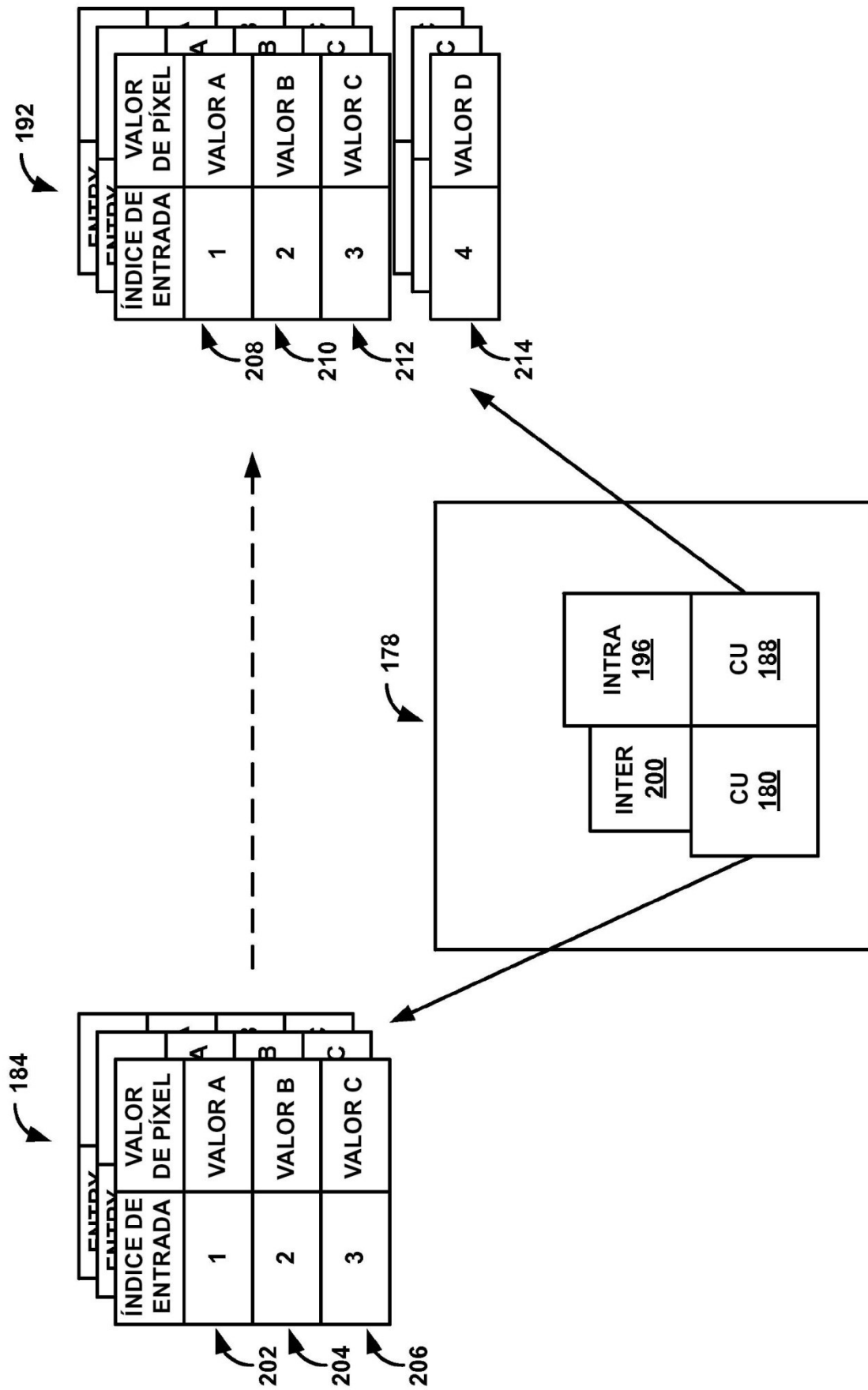


FIG. 4

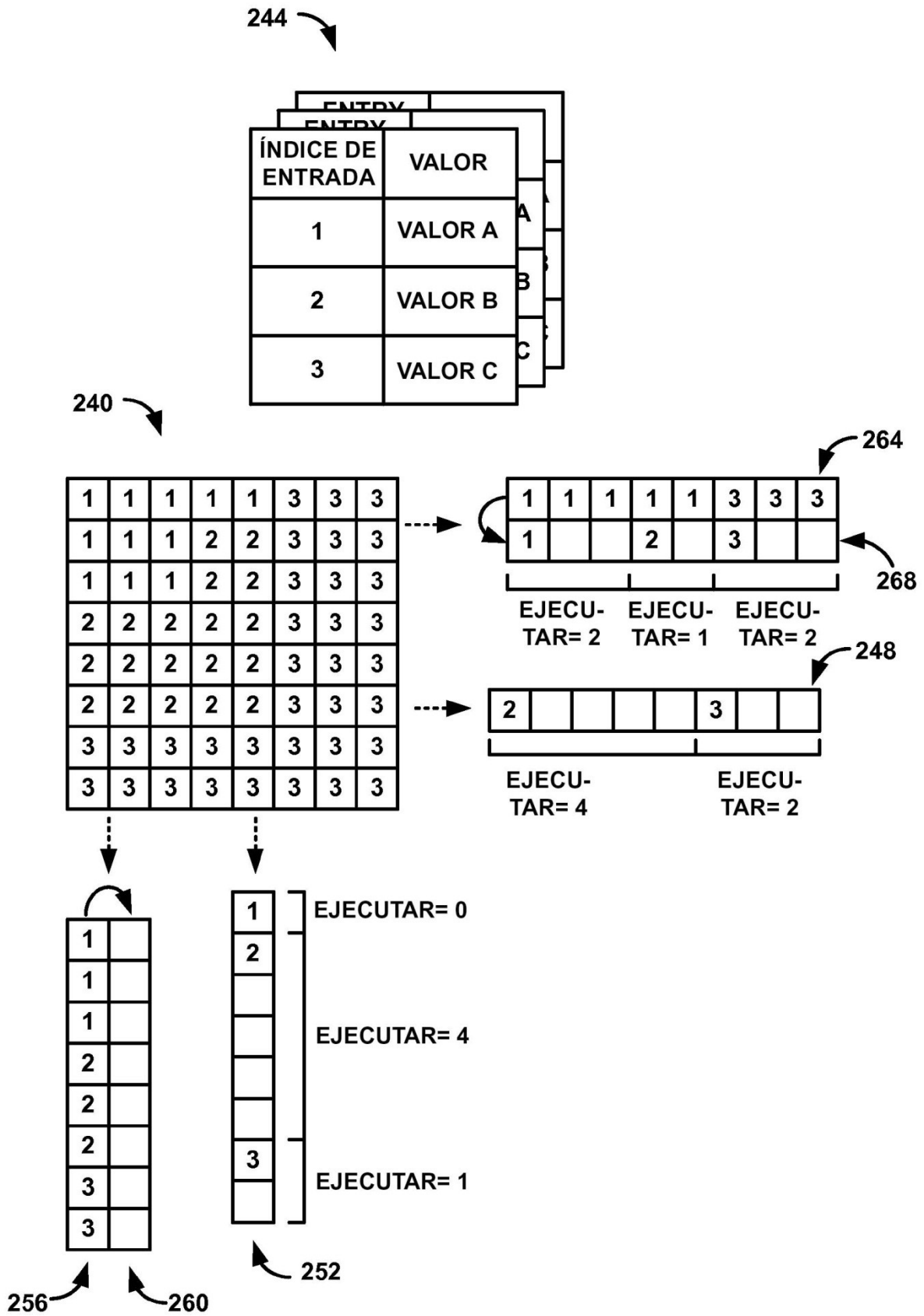


FIG. 5

○ MUESTRA DE LUMA ⊗ MUESTRA DE CROMA

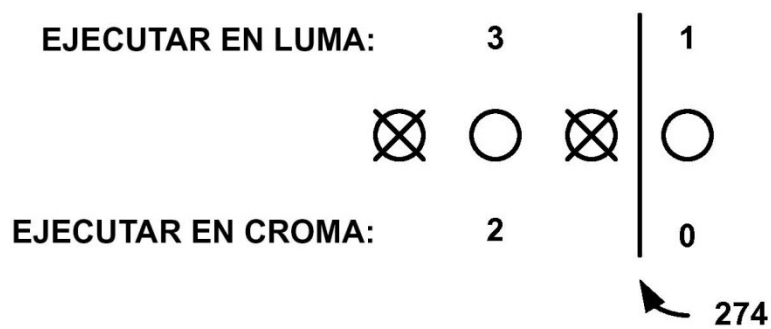
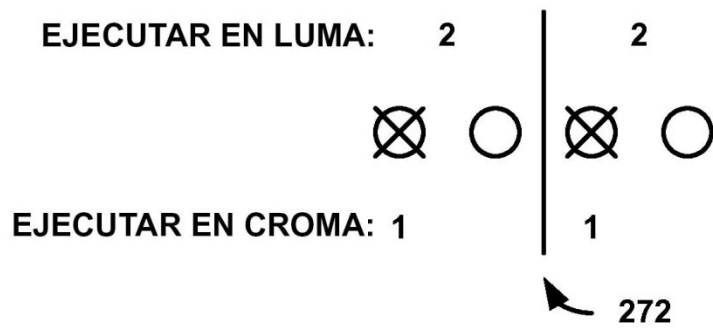
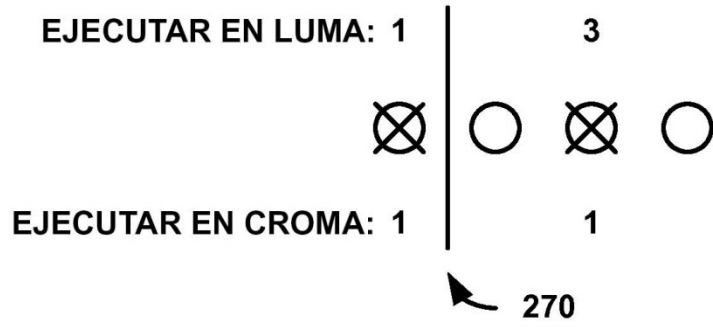


FIG. 6

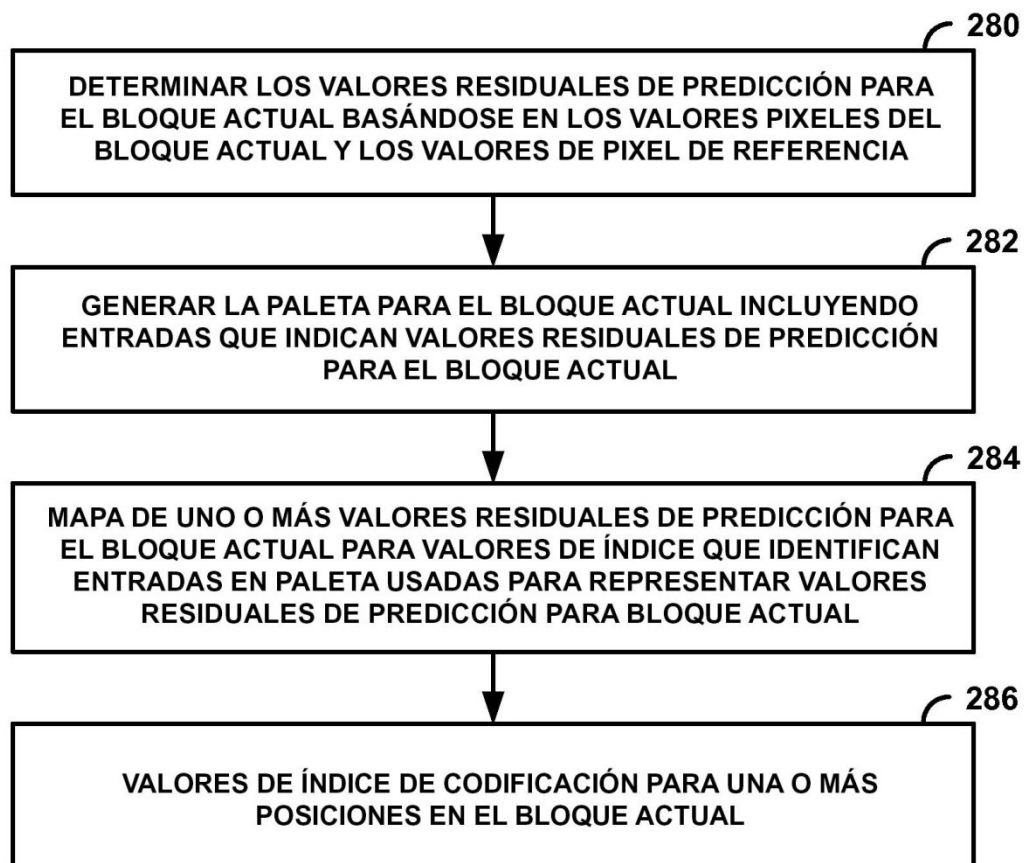


FIG. 7

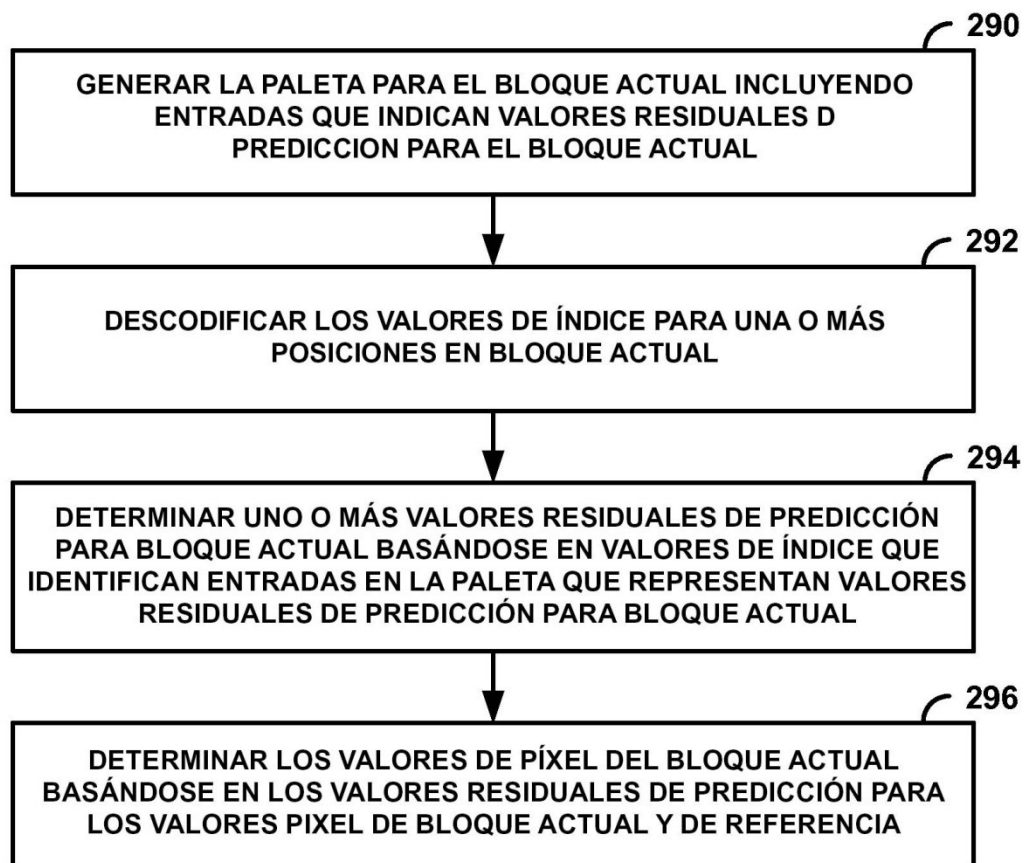


FIG. 8

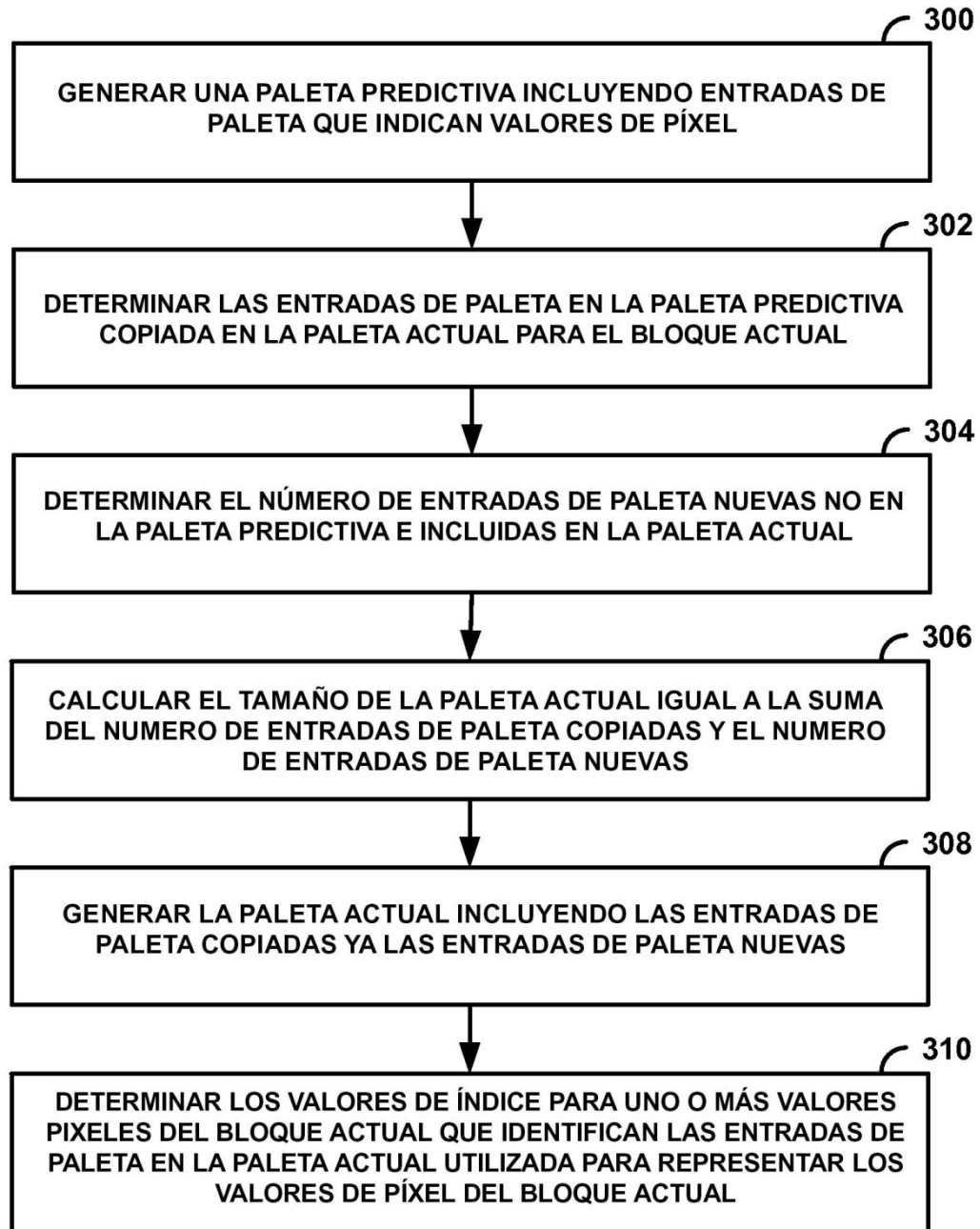


FIG. 9