

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 751 576**

51 Int. Cl.:

<b>H04N 19/70</b>	(2014.01)
<b>H04N 19/13</b>	(2014.01)
<b>H04N 19/91</b>	(2014.01)
<b>H04N 19/184</b>	(2014.01)
<b>H04N 19/176</b>	(2014.01)
<b>H04N 19/186</b>	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **09.06.2016 PCT/US2016/036572**

87 Fecha y número de publicación internacional: **15.12.2016 WO16201032**

96 Fecha de presentación y número de la solicitud europea: **09.06.2016 E 16730666 (1)**

97 Fecha y número de publicación de la concesión europea: **24.07.2019 EP 3308542**

54 Título: **Agrupación de períodos de derivación de paleta para codificación de vídeo**

30 Prioridad:

**12.06.2015 US 201562175137 P**  
**08.06.2016 US 201615177201**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**01.04.2020**

73 Titular/es:

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

72 Inventor/es:

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

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

**ES 2 751 576 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**

Agrupación de períodos de derivación de paleta para codificación de vídeo

5 **CAMPO TÉCNICO**

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

**ANTECEDENTES**

10 **[0002]** Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, que incluye 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 continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, codificación avanzada de vídeo (AVC), ITU-T-H.265, el estándar de codificación de vídeo de alta eficacia (HEVC) y las extensiones de dichos estándares. 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 **[0003]** Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo (es decir, una trama de vídeo o una parte de una trama de vídeo) se puede dividir en bloques de vídeo. Los bloques de vídeo de un segmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma imagen. Los bloques de vídeo de un segmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma imagen, o predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes se pueden denominar tramas y las imágenes de referencia se pueden denominar tramas de referencia.

25 **[0004]** La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales se pueden transformar desde el dominio del píxel hasta un dominio de transformada, dando como resultado coeficientes residuales, que a continuación se pueden cuantificar. Los coeficientes cuantificados, dispuestos inicialmente en una matriz bidimensional, se pueden explorar a fin de generar un vector unidimensional de coeficientes, y se puede aplicar codificación de entropía para lograr incluso más compresión.

**SUMARIO**

45 **[0005]** En un ejemplo, un procedimiento de decodificación de datos de vídeo incluye decodificar, a partir de una corriente de bits de vídeo codificada, un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para un bloque actual de datos de vídeo; decodificar, a partir de la corriente de bits de vídeo codificada y en una posición de la corriente de bits de vídeo codificada que está después del elemento sintáctico que indica si el proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo, uno o más elementos sintácticos relacionados con un parámetro de cuantificación (QP) delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo; y decodificar el bloque actual de datos de vídeo en base a la paleta para el bloque actual de datos de vídeo y el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo.

55 **[0006]** En otro ejemplo, un procedimiento de codificación de datos de vídeo incluye codificar, en una corriente de bits de vídeo codificada, un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para un bloque actual de datos de vídeo; codificar, en la corriente de bits de vídeo codificada y en una posición de la corriente de bits de vídeo codificada que está después del elemento sintáctico que indica si el proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo, uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo; y codificar el bloque actual de datos de vídeo en base a la paleta para el bloque actual de datos de vídeo y el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo.

65

5 [0007] En otro ejemplo, un dispositivo para codificar datos de vídeo incluye una memoria configurada para almacenar datos de vídeo y uno o más procesadores. En este ejemplo, el uno o más procesadores están configurados para: codificar, en una corriente de bits de vídeo codificada, un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para un bloque actual de datos de vídeo; codificar, en la corriente de bits de vídeo codificada y en una posición de la corriente de bits de vídeo codificada que está después del elemento sintáctico que indica si el proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo, uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo; y codificar el bloque actual de datos de vídeo en base a la paleta para el bloque actual de datos de vídeo y el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo.

15 [0008] En otro ejemplo, un dispositivo para codificar datos de vídeo incluye medios para codificar, en una corriente de bits de vídeo codificada, un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para un bloque actual de datos de vídeo; medios para codificar, en la corriente de bits de vídeo codificada y en una posición de la corriente de bits de vídeo codificada que está después del elemento sintáctico que indica si el proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo, uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo; y medios para codificar el bloque actual de datos de vídeo en base a la paleta para el bloque actual de datos de vídeo y el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo.

25 [0009] En otro ejemplo, un medio de almacenamiento legible por ordenador almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo de codificación de vídeo: codifiquen, en una corriente de bits de vídeo codificada, un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para un bloque actual de datos de vídeo; codifiquen, en la corriente de bits de vídeo codificada y en una posición de la corriente de bits de vídeo codificada que está después del elemento sintáctico que indica si el proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo, uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo; y codifiquen el bloque actual de datos de vídeo en base a la paleta para el bloque actual de datos de vídeo y el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo.

35 [0010] En otro ejemplo, un medio de almacenamiento legible por ordenador almacena al menos una parte de una corriente de bits de vídeo codificada que, cuando se procesa mediante un dispositivo de descodificación de vídeo, hace que uno o más procesadores del dispositivo de descodificación de vídeo: determinen si se aplica un proceso de transposición a unos índices de paleta de una paleta para un bloque actual de datos de vídeo; y descodifiquen el bloque actual de los datos de vídeo en base a la paleta para el bloque actual de datos de vídeo y un QP delta y uno o más desplazamientos de QP de croma para el bloque actual de datos de vídeo, en el que uno o más elementos sintácticos relacionados con el QP delta y uno o más elementos sintácticos relacionados con el uno o más desplazamientos de QP de croma para el bloque actual de datos de vídeo están localizados en una posición de la corriente de bits de vídeo codificada que está después de un elemento sintáctico que indica si el proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo.

45 [0011] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones. La divulgación suficiente para la invención se encuentra en el modo de realización de la Tabla 4.

**BREVE DESCRIPCIÓN DE LOS DIBUJOS**

50 [0012]

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

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

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

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

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

La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de proceso para descodificar un bloque de datos de vídeo usando un modo de paleta, de acuerdo con una o más técnicas de esta divulgación.

La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de proceso para codificar un bloque de datos de vídeo usando un modo de paleta, de acuerdo con una o más técnicas de esta divulgación.

## DESCRIPCIÓN DETALLADA

[0013] Esta divulgación describe técnicas para codificación y compresión de vídeo. En particular, esta divulgación describe técnicas para codificación basada en paleta de datos de vídeo. Por ejemplo, esta divulgación describe técnicas para admitir la codificación de contenido de vídeo, especialmente contenido de pantalla con codificación de paleta, tales como técnicas para mejorar la binarización de índices de paleta y técnicas para señalar para codificación de paleta.

[0014] En la codificación de vídeo tradicional, se supone que las imágenes son de tono continuo y espacialmente homogéneas. En base a estas suposiciones, se han desarrollado diversas herramientas, tales como la transformada basada en bloques, el filtrado, etc., y dichas herramientas han demostrado un buen rendimiento para vídeos de contenido natural.

[0015] Sin embargo, en aplicaciones como las de escritorio remoto, trabajo colaborativo y visualización inalámbrica, el contenido de pantalla generado por ordenador puede ser el contenido dominante que se debe comprimir. Este tipo de contenido tiende a tener tonos discretos y líneas características nítidas y límites de objetos de alto contraste. Es posible que el supuesto de tono continuo y homogeneidad ya no se aplique y, por tanto, las técnicas de codificación de vídeo tradicionales pueden no ser formas eficaces para comprimir.

[0016] En base en las características del vídeo de contenido de pantalla, se introduce la codificación de paleta para mejorar la eficacia de codificación de contenido de pantalla (SCC) tal como se propone en el documento de Guo *et al.*, "Palette Mode for Screen Content Coding", Equipo Mixto de Colaboración sobre Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 13.<sup>a</sup> reunión: Incheon, KR, 18-26 abr. de 2013, documento: JCTVC-M0323, disponible en [http://phenix.itsudparis.eu/jct/doc\\_end\\_user/documents/13\\_Incheon/wg11/JCTVC-M0323-v3.zip](http://phenix.itsudparis.eu/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0323-v3.zip), ("JCTVC-M0323" más adelante en el presente documento). Específicamente, la codificación de paleta introduce una tabla de consulta, es decir, una paleta de colores, para comprimir valores de píxel repetitivos en base al hecho de que en SCC, los colores de dentro de una CU normalmente se concentran en unos pocos valores máximos. Dada una paleta para una CU específica, los píxeles de dentro de la CU se mapean a índices de paleta. En la segunda fase, se propone una copia efectiva del procedimiento de longitud de recorrido izquierdo para comprimir eficazmente el patrón repetitivo del bloque de índices. En algunos ejemplos, el modo de codificación de índices de paleta se puede generalizar tanto para copiar desde la izquierda como para copiar desde arriba con codificación por longitud de recorrido. Debe tenerse en cuenta que, en algunos ejemplos, puede ser que no se invoque ningún proceso de transformación para la codificación de paleta para evitar una disminución de definición de los bordes nítidos, lo cual podría tener un enorme impacto negativo en la calidad visual del contenido de pantalla.

[0017] Como se analiza anteriormente, esta divulgación describe la codificación basada en paleta, que puede ser adecuada en particular para la codificación de contenido generado por pantalla. Por ejemplo, se va a suponer que un área particular de datos de vídeo tiene un número relativamente pequeño de colores. Un codificador de vídeo (un codificador de vídeo o un descodificador de vídeo) puede codificar una denominada "paleta" como una tabla de colores para representar los datos de vídeo del área particular (por ejemplo, un bloque dado). Cada píxel se puede asociar con una entrada de la paleta que representa el color del píxel. Por ejemplo, el codificador de vídeo puede codificar un índice que mapea el valor de píxel al valor adecuado de la paleta.

[0018] En el ejemplo anterior, un codificador de vídeo puede codificar un bloque de datos de vídeo determinando una paleta para el bloque, localizando una entrada en la paleta para representar el valor de color de cada píxel y codificando la paleta con valores de índice para los píxeles mapeando el valor de píxel a la paleta. Un descodificador de vídeo puede obtener, a partir de una corriente de bits codificada, una paleta para un bloque, así como valores de índices para los píxeles del bloque. El descodificador de vídeo puede mapear los valores de índice de los píxeles a unas entradas de la paleta para reconstruir los diversos valores de píxel de luma y croma del bloque.

[0019] El ejemplo anterior pretende proporcionar una descripción general de la codificación basada en paleta. En diversos ejemplos, las técnicas descritas en esta divulgación pueden incluir técnicas para diversas combinaciones de una o más de una señalización de modos de codificación basados en paleta, una transmisión de paletas, una predicción de paletas, una obtención de paletas y una transmisión de mapas de codificación basados en paleta y otros elementos sintácticos. Dichas técnicas pueden mejorar la eficacia de la codificación de vídeo, por ejemplo, requerir menos bits para representar contenido generado por pantalla.

[0020] Por ejemplo, de acuerdo con unos aspectos de esta divulgación, un codificador de vídeo (codificador de vídeo o descodificador de vídeo) puede codificar uno o más elementos sintácticos para cada bloque que se codifica usando un modo de codificación de paleta. Por ejemplo, el codificador de vídeo puede codificar una **palette\_mode\_flag** para

indicar si se va a usar un modo de codificación basada en paleta para codificar un bloque en particular. En este ejemplo, un codificador de vídeo puede codificar una **palette\_mode\_flag** con un valor que es igual a uno, para especificar que el bloque que se está codificando actualmente ("bloque actual") se codifica usando un modo de paleta. En este caso, un descodificador de vídeo puede obtener la **palette\_mode\_flag** a partir de la corriente de bits codificada y aplicar el modo de codificación basada en paleta para descodificar el bloque. En casos en los que hay disponible más de un modo de codificación basada en paleta (por ejemplo, hay disponible más de una técnica basada en paleta para codificación), uno o más elementos sintácticos pueden indicar uno de una pluralidad de modos de paleta diferentes para el bloque.

**[0021]** En algunos casos, el codificador de vídeo puede codificar una **palette\_mode\_flag** con un valor 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 puede codificar el bloque usando cualquiera de una variedad de modos de codificación interpredictiva, intrapredictiva u otros. Cuando la **palette\_mode\_flag** es igual a cero, el codificador de vídeo puede codificar 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. El uso de la **palette\_mode\_flag** se describe con propósitos ejemplificativos. En otros ejemplos, otros elementos sintácticos, tales como códigos multibit, se pueden usar para indicar si el modo de codificación basado en paleta se va a usar para uno o más bloques, o para indicar cuáles, de una pluralidad de modos, se van a usar.

**[0022]** Cuando se usa un modo de codificación basada en paleta, un codificador puede transmitir una paleta en la corriente de bits de datos de vídeo codificada para su uso por un descodificador. Una paleta se puede transmitir para cada bloque o compartir entre un número de bloques de una imagen o segmento. La paleta se puede referir a un número de valores de píxel que son dominantes y/o representativos para el bloque, incluyendo, por ejemplo, un valor de luma y dos valores de croma.

**[0023]** En algunos ejemplos, un elemento sintáctico, tal como una marca de transposición, se puede codificar para indicar si un proceso de transposición se aplica a unos índices de paleta de una paleta actual. Si la marca de transposición es cero, los índices de paleta para unas muestras se pueden codificar en una exploración transversal horizontal. De forma similar, si la marca de transposición es uno, los índices de paleta para unas muestras se pueden codificar en una exploración transversal vertical. Se puede considerar que esto significa descodificar los valores de índice suponiendo una exploración transversal horizontal y a continuación una transposición del bloque (de filas a columnas).

**[0024]** Los aspectos de esta divulgación incluyen técnicas para codificar la paleta. Por ejemplo, de acuerdo con unos aspectos de esta divulgación, un codificador de vídeo puede codificar uno o más elementos sintácticos para definir una paleta. Algunos ejemplos de elementos sintácticos que un codificador de vídeo puede codificar para definir una paleta actual para un bloque actual de datos de vídeo incluyen, pero no se limitan a, un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de la paleta actual (p. ej., **palette\_transpose\_flag**) (es decir, si el uno o más elementos sintácticos relacionados con el parámetro de cuantificación (QP) delta (p. ej., **cu\_qp\_delta\_palette\_abs**, **cu\_qp\_delta\_palette\_sign\_flag**, **cu\_chroma\_qp\_palette\_offset\_flag**, y/o **cu\_chroma\_qp\_palette\_offset\_idx**), uno o más elementos sintácticos relacionados con unos desplazamientos de QP de croma para el bloque actual de datos de vídeo, uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si unas entradas de una paleta predictora se reúsan en la paleta actual (por ejemplo, **palette\_predictor\_run**), uno o más elementos sintácticos que indican un número de entradas de la paleta actual que se señalan explícitamente (por ejemplo, **num\_signalled\_palette\_entries**), uno o más elementos sintácticos que indican un valor de un componente de una entrada de paleta de la paleta actual (por ejemplo, **palette\_entry**), uno o más elementos sintácticos que indican si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape (por ejemplo, **palette\_escape\_val\_present\_flag**), uno o más elementos sintácticos que indican un número de entradas de la paleta actual que se señalan o deducen explícitamente (por ejemplo, **num\_palette\_indices\_idc**), y uno o más elementos sintácticos que indican índices de una matriz de entradas de paleta actual (p. ej., **palette\_index\_idc**). Por ejemplo, cuando se funciona de acuerdo con el borrador 3 de codificación de contenido de pantalla (SCC) HEVC (Joshi *et al.*, "High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 3", Equipo Mixto de Colaboración sobre Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 20.<sup>a</sup> reunión: Ginebra, CH, 10 de febrero - 17 de febrero de 2015, documento: JCTVC-T1005, disponible en [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/20\\_Geneva/wg11/JCTVC-T1005-v2.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/20_Geneva/wg11/JCTVC-T1005-v2.zip), ("borrador 3 de SCC HEVC" más adelante en el presente documento), un codificador de vídeo puede señalar los elementos sintácticos enumerados en la tabla de sintaxis `palette_coding()` (sección 7.3.8.8 del borrador 3 de SCC HEVC), que se reproduce a continuación como Tabla 1.

Tabla 1

<code>palette_coding(x0, y0, nCbS) {</code>	Descriptor
<code>palettePredictionFinished = 0</code>	
<code>NumPredictedPaletteEntries = 0</code>	

	Descriptor
palette_coding(x0, y0, nCbS) {	
for(i = 0; i < PredictorPaletteSize && !palettePredictionFinished && NumPredictedPaletteEntries < palette_max_size; i++) {	
<b>palette_predictor_run</b>	ue(v)
if(palette_predictor_run != 1) {	
if(palette_predictor_run > 1)	
i += palette_predictor_run - 1	
PalettePredictorEntryReuseFlag[i] = 1	
NumPredictedPaletteEntries++	
} else	
palettePredictionFinished = 1	
}	
if(NumPredictedPaletteEntries < palette_max_size)	
<b>num_signalled_palette_entries</b>	ue(v)
numComps = (ChromaArrayType == 0) ? 1 : 3	
for(cldx = 0; cldx < numComps; cldx++)	
for(i = 0; i < num_signalled_palette_entries; i++)	
<b>palette_entry</b>	ae(v)
if(CurrentPaletteSize != 0)	
<b>palette_escape_val_present_flag</b>	ae(v)
if(palette_escape_val_present_flag) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
<b>cu_qp_delta_palette_abs</b>	ae(v)
if(cu_qp_delta_palette_abs)	
<b>cu_qp_delta_palette_sign_flag</b>	ae(v)
}	
if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
<b>cu_chroma_qp_palette_offset_flag</b>	ae(v)
if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
<b>cu_chroma_qp_palette_offset_idx</b>	ae(v)
}	
}	
}	
if(MaxPaletteIndex > 0) {	
<b>palette_transpose_flag</b>	ae(v)
<b>num_palette_indices_idc</b>	ae(v)
for(i=0; i < NumPaletteIndices; i++) {	
<b>palette_index_idc</b>	ae(v)
PaletteIndexIdc[i] = palette_index_idc	
}	
<b>last_palette_run_type_flag</b>	ae(v)
}	
CurrNumIndices = 0	
PaletteScanPos = 0	

ES 2 751 576 T3

	Descriptor
palette_coding(x0, y0, nCbS) {	
while(PaletteScanPos < nCbS * nCbS) {	
xC = x0 + travScan[PaletteScanPos][0]	
yC = y0 + travScan[PaletteScanPos][1]	
if(PaletteScanPos > 0) {	
xcPrev = x0 + travScan[PaletteScanPos - 1][0]	
ycPrev = y0 + travScan[PaletteScanPos - 1][1]	
}	
PaletteRun = nCbS * nCbS - PaletteScanPos - 1	
if(MaxPaletteIndex > 0 && CurrNumIndices < NumPaletteIndices) {	
if(PaletteScanPos >= nCbS && palette_run_type_flag[xcPrev][ycPrev] != COPY_ABOVE_MODE && PaletteScanPos < nCbS * nCbS - 1) {	
<b>palette_run_type_flag[xC][yC]</b>	ae(v)
}	
readIndex = 0	
if(palette_run_type_flag[xC][yC] == COPY_INDEX_MODE && AdjustedMaxPaletteIndex > 0)	
readIndex = 1	
maxPaletteRun = nCbS * nCbS - PaletteScanPos - 1	
if(AdjustedMaxPaletteIndex > 0 && ((CurrNumIndices + readIndex) < NumPaletteIndices    palette_run_type_flag[xC][yC] != last_palette_run_type_flag))	
if(maxPaletteRun > 0) {	
<b>palette_run_msb_id_plus1</b>	ae(v)
if(palette_run_msb_id_plus1 > 1)	
<b>palette_run_refinement_bits</b>	ae(v)
}	
CurrNumIndices += readIndex	
}	
runPos = 0	
while (runPos <= paletteRun) {	
xR = x0 + travScan[PaletteScanPos][0]	
yR = y0 + travScan[PaletteScanPos][1]	
if(palette_run_type_flag[xC][yC] == COPY_INDEX_MODE) {	
PaletteSampleMode[xR][yR] = COPY_INDEX_MODE	
PaletteIndexMap[xR][yR] = CurrPaletteIndex	
} else {	
PaletteSampleMode[xR][yR] = COPY_ABOVE_MODE	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xR][yR - 1]	
}	
runPos++	
PaletteScanPos++	
}	
}	

palette_coding(x0, y0, nCbS) {	<b>Descriptor</b>
if(palette_escape_val_present_flag) {	
sPos = 0	
while(sPos < nCbS * nCbS) {	
xC = x0 + travScan[sPos][0]	
yC = y0 + travScan[sPos][1]	
if(PaletteIndexMap[xC][yC] == MaxPaletteIndex) {	
for(cldx = 0; cldx < numComps; cldx++)	
if(cldx == 0    (xR % 2 == 0 && yR % 2 == 0 && ChromaArrayType == 1)    (xR % 2 == 0 && ChromaArrayType == 2)    ChromaArrayType == 3) {	
<b>palette_escape_val</b>	ae(v)
PaletteEscapeVal[cldx][xC][yC] = palette_escape_val	
}	
}	
sPos++	
}	
}	

**[0025]** Además de proporcionar un orden en el que los elementos sintácticos se incluyen en una corriente de bits, la Tabla 1 también proporciona un descriptor para cada uno de los elementos sintácticos que indica un tipo de codificación para cada elemento sintáctico. Como ejemplo, un codificador de vídeo puede codificar elementos sintácticos con el descriptor ue(v) usando códigos Golomb exponenciales enteros sin signo de orden 0 con el bit izquierdo en primer lugar. Como otro ejemplo, un codificador de vídeo puede codificar elementos sintácticos con el descriptor ae(v) usando códigos aritméticos de entropía adaptativos al contexto (CABAC). Cuando se codifican períodos de un elemento sintáctico usando CABAC, un codificador de vídeo puede codificar uno o más de los períodos usando un contexto y/o puede codificar uno o más de los períodos sin contexto. La codificación de un período usando CABAC sin contexto se puede denominar modo de derivación. El borrador 3 de SCC HEVC proporciona además una tabla (Tabla 9-47 del borrador 3 de SCC HEVC), parcialmente reproducida a continuación como la Tabla 2, que indica qué períodos de los elementos sintácticos enumerados en la Tabla 1 se codifican con contextos (es decir, como se indica mediante contexto "0" y contexto "1") y qué períodos se codifican en modo de derivación.

**Tabla 2**

Elemento sintáctico	Ind. per.					
	0	1	2	3	4	>= 5
palette_predictor_run	derivación	derivación	derivación	derivación	derivación	derivación
num_signalled_palette_entries	derivación	derivación	derivación	derivación	derivación	derivación
palette_entry	derivación	derivación	derivación	derivación	derivación	derivación
palette_escape_val_present_flag	derivación	n.d.	n.d.	n.d.	n.d.	n.d.
cu_qp_delta_palette_abs	0	1	1	1	1	derivación
cu_qp_delta_palette_sign_flag	derivación	n.d.	n.d.	n.d.	n.d.	n.d.
cu_chroma_qp_palette_offset_flag	0	n.d.	n.d.	n.d.	n.d.	n.d.
cu_chroma_qp_palette_offset_idx	0	0	0	0	0	n.d.
palette_transpose_flag	0	n.d.	n.d.	n.d.	n.d.	n.d.
num_palette_indices_idc	derivación	derivación	derivación	derivación	derivación	derivación
last_palette_run_type_flag	0	n.d.	n.d.	n.d.	n.d.	n.d.
palette_run_type_flag	0	n.d.	n.d.	n.d.	n.d.	n.d.



Elemento sintáctico	Ind. per.					
	0	1	2	3	4	>= 5
palette_index_idc	derivación	derivación	derivación	derivación	derivación	derivación
palette_run_msb_id_plus1	(cláusula 9.3.4.2.8)					
palette_run_refinement_bits	derivación	derivación	derivación	derivación	derivación	derivación
palette_escape_val	derivación	derivación	derivación	derivación	derivación	derivación

5 [0026] Una comparación de la Tabla 1 y la Tabla 2 muestra que el borrador 3 de SCC HEVC recomienda que todos los elementos sintácticos antes de **cu\_qp\_delta\_palette\_abs** (es decir, **num\_signalled\_palette\_entries**, **palette\_entry** y **palette\_escape\_val\_present\_flag**) se codifiquen en derivación. De forma similar, los elementos sintácticos después de **palette\_transpose\_flag** y antes de **last\_palette\_run\_type\_flag** (es decir, **num\_palette\_indices\_idc** y **palette\_index\_idc**) también se codifican en derivación.

10 [0027] Al codificar un período usando CABAC con un contexto, un codificador de vídeo puede cargar el contexto desde el almacenamiento hasta la memoria. En algunos ejemplos, un codificador de vídeo puede tener recursos de memoria limitados disponibles, y/o la carga de un contexto en memoria puede llevar mucho tiempo. Así pues, puede ser deseable que un codificador de vídeo reduzca al mínimo la cantidad de veces que los contextos se cargan en memoria. En algunos ejemplos, agrupar períodos de derivación conjuntamente puede reducir la cantidad de veces que los contextos se cargan en memoria, lo que puede incrementar el rendimiento CABAC.

15 [0028] En el documento de Ye *et al.*, "CE1-related: Palette Mode Context and Codeword Simplification", Equipo Mixto de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 21.<sup>a</sup> reunión: Varsovia, PL, 19-26 de junio de 2015, documento: JCTVC-U0090, disponible en [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/documents/21\\_Warsaw/wg11/JCTVC-U0090-v1.zip](http://phenix.it-sudparis.eu/jct/doc_end_user/documents/21_Warsaw/wg11/JCTVC-U0090-v1.zip) ("JCTVC-U0090" más adelante en el presente documento), se propuso que la **palette\_transpose\_flag** se señalizara después de **last\_palette\_run\_type\_flag**. Específicamente, el documento JCTVC-U0090 propone modificar la tabla de sintaxis `palette_coding()` como se muestra a continuación en la Tabla 3 (donde se inserta texto en *cursiva* y se suprime texto en *[[cursiva entre corchetes dobles]]*).

Tabla 3

if(MaxPaletteIndex > 0) {	
<b>[[palette_transpose_flag]]</b>	<b>[[ae(v)]]</b>
<b>num_palette_indices_idc</b>	<b>ae(v)</b>
for(i=0; i < NumPaletteIndices; i++) {	
<b>palette_index_idc</b>	<b>ae(v)</b>
PaletteIndexIdc[i] = palette_index_idc	
}	
<b>last_palette_run_type_flag</b>	<b>ae(v)</b>
<b>palette_transpose_flag</b>	<b>ae(v)</b>
}	

25 [0029] Sin embargo, en algunos ejemplos, la disposición de elementos sintácticos propuesta por JCTVC-U0090 puede no ser óptima. Por ejemplo, cuando unos elementos sintácticos relacionados con un QP delta (es decir, **cu\_qp\_delta\_palette\_abs** y **cu\_qp\_delta\_palette\_sign\_flag**) y un desplazamiento de QP de croma (es decir, **cu\_chroma\_qp\_palette\_offset\_flag** y **cu\_chroma\_qp\_palette\_offset\_idx**) están presentes, la disposición de elementos sintácticos propuesta por JCTVC-U0090 puede no resultar en un agrupamiento de ningún período de derivación adicional.

30 [0030] De acuerdo con una o más técnicas de esta divulgación, un codificador de vídeo puede codificar los elementos sintácticos usados para definir una paleta actual, de modo que unos elementos sintácticos que se codifican usando un modo de derivación se codifican consecutivamente. Por ejemplo, en lugar de codificar uno o más elementos sintácticos relacionados con un parámetro de cuantificación (QP) delta y/o unos desplazamientos de QP de croma para un bloque actual de datos de vídeo antes de un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para el bloque actual de datos de vídeo, un codificador de vídeo puede codificar el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo después del elemento sintáctico que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de vídeo.

**[0031]** Un ejemplo de cómo se puede modificar la tabla de sintaxis `palette_coding()` para desplazar la señalización de los elementos sintácticos relacionados con un QP delta y unos desplazamientos de QP de croma después de la `palette_transpose_flag` se muestra a continuación en la Tabla 4 (donde se inserta texto en *cursiva* y se suprime texto en *[[cursiva entre corchetes dobles]]* en relación con una versión previa de la Tabla 4 en el borrador 3 de SCC HEVC).

5

Tabla 4

	Descriptor
<code>palette_coding(x0, y0, nCbS) {</code>	
<code>palettePredictionFinished = 0</code>	
<code>NumPredictedPaletteEntries = 0</code>	
<code>for(i = 0; i &lt; PredictorPaletteSize &amp;&amp; !palettePredictionFinished &amp;&amp; NumPredictedPaletteEntries &lt; palette_max_size; i++) {</code>	
<b><code>palette_predictor_run</code></b>	<code>ue(v)</code>
<code>if(palette_predictor_run != 1) {</code>	
<code>if(palette_predictor_run &gt; 1)</code>	
<code>i += palette_predictor_run - 1</code>	
<code>PalettePredictorEntryReuseFlag[i] = 1</code>	
<code>NumPredictedPaletteEntries++</code>	
<code>} else</code>	
<code>palettePredictionFinished = 1</code>	
<code>}</code>	
<code>if(NumPredictedPaletteEntries &lt; palette_max_size)</code>	
<b><code>num_signalled_palette_entries</code></b>	<code>ue(v)</code>
<code>numComps = (ChromaArrayType == 0) ? 1 : 3</code>	
<code>for(cldx = 0; cldx &lt; numComps; cldx++)</code>	
<code>for(i = 0; i &lt; num_signalled_palette_entries; i++)</code>	
<b><code>palette_entry</code></b>	<code>ae(v)</code>
<code>if(CurrentPaletteSize != 0)</code>	
<b><code>palette_escape_val_present_flag</code></b>	<code>ae(v)</code>
<code>[[if(palette_escape_val_present_flag) {}]]</code>	
<code>[[if(cu_qp_delta_enabled_flag &amp;&amp; !IsCuQpDeltaCoded) {}]]</code>	
<b><code>[[cu_qp_delta_palette_abs]]</code></b>	<code>[[ae(v)]]</code>
<code>[[if(cu_qp_delta_palette_abs)]]</code>	
<b><code>[[cu_qp_delta_palette_sign_flag]]</code></b>	<code>[[ae(v)]]</code>
<code>[[{}]]</code>	
<code>[[if(cu_chroma_qp_offset_enabled_flag &amp;&amp; !IsCuChromaQpOffsetCoded) {}]]</code>	
<b><code>[[cu_chroma_qp_palette_offset_ag]]</code></b>	<code>[[ae(v)]]</code>
<code>[[if(cu_chroma_qp_offset_flag &amp;&amp; chroma_qp_offset_list_len_minus1 &gt; 0)]]</code>	
<b><code>[[cu_chroma_qp_palette_offset_idx]]</code></b>	<code>[[ae(v)]]</code>
<code>[[{}]]</code>	
<code>[[{}]]</code>	
<code>if(MaxPaletteIndex &gt; 0) {</code>	
<b><code>[[palette_transpose_flag]]</code></b>	<code>[[ae(v)]]</code>
<b><code>num_palette_indices_idc</code></b>	<code>ae(v)</code>
<code>for(i=0; i &lt; NumPaletteIndices; i++) {</code>	
<b><code>palette_index_idc</code></b>	<code>ae(v)</code>

	Descriptor
palette_coding(x0, y0, nCbS) {	
PaletteIndexIdc[i] = palette_index_idc	
}	
<b>last_palette_run_type_flag</b>	ae(v)
<i>palette_transpose_flag</i>	ae(v)
}	
if( <i>palette_escape_val_present_flag</i> ) {	
if( <i>cu_qp_delta_enabled_flag</i> && ! <i>IsCuQpDeltaCoded</i> ) {	
<b>cu_qp_delta_palette_abs</b>	ae(v)
if( <i>cu_qp_delta_palette_abs</i> )	
<b>cu_qp_delta_palette_sign_flag</b>	ae(v)
}	
if( <i>cu_chroma_qp_offset_enabled_fag</i> && ! <i>IsCuChromaQpOffsetcoded</i> ) {	
<b>cu_chroma_qp_palette_offset_flag</b>	ae(v)
if( <i>cu_chroma_qp_offset_flag</i> && <i>chroma_qp_offset_list_len_minus1</i> > 0)	
<b>cu_chroma_qp_palette_offset_idx</b>	ae(v)
}	
}	
}	
CurrNumIndices = 0	
PaletteScanPos = 0	
...	

[0032] Desplazando uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo después del elemento sintáctico que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de vídeo, el codificador de vídeo puede agrupar (es decir, codificar consecutivamente) un número mayor de elementos sintácticos que se codifican usando el modo de derivación. Por ejemplo, desplazando el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo después del elemento sintáctico que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de vídeo, el codificador de vídeo puede agrupar uno o más elementos sintácticos que indican un número de entradas de la paleta actual que se señalan o deducen explícitamente (por ejemplo, **num\_palette\_indices\_idc**) y uno o más elementos sintácticos que indican unos índices de una matriz de entradas de paleta actual (por ejemplo, **palette\_index\_idc**) con uno o más elementos sintácticos relacionados con unos desplazamientos de QP de croma para el bloque actual de datos de vídeo, uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si las entradas de una paleta predictora se reúsan en la paleta actual (por ejemplo, **palette\_predictor\_run**), uno o más elementos sintácticos que indican un número de entradas de la paleta actual que se señalan explícitamente (por ejemplo, **num\_signalled\_palette\_entries**), uno o más elementos sintácticos que indican un valor de un componente de una entrada de paleta de la paleta actual (por ejemplo, **palette\_entry**), y uno o más elementos sintácticos que indican si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape (por ejemplo, **palette\_escape\_val\_present\_flag**). De esta manera, las técnicas de esta divulgación pueden incrementar el rendimiento CABAC, lo que puede reducir el tiempo necesario para codificar datos de vídeo usando la codificación de modo de paleta. Por ejemplo, agrupando los elementos sintácticos codificados en derivación, un codificador de vídeo puede codificar secuencialmente los elementos sintácticos agrupados usando sin arrancar, detener, volver a arrancar, recargar y restablecer un motor de codificación CABAC

[0033] La Tabla 4 es solo un ejemplo de cómo se pueden disponer los elementos sintácticos. En algunos ejemplos, los elementos sintácticos relacionados con un QP delta y un desplazamiento de QP de croma se pueden desplazar más abajo por la tabla de sintaxis. Por ejemplo, los elementos sintácticos relacionados con el un QP delta y un desplazamiento de QP de croma se podrían colocar justo antes de los valores de componente para muestras de escape (es decir, **palette\_escape\_val**). Un ejemplo de cómo los elementos sintácticos relacionados con un QP delta y un desplazamiento de QP de croma se podrían colocar justo antes de los valores de componente para unas muestras de escape se muestra a continuación en la Tabla 5 (donde se inserta texto en *cursiva* y se suprime texto en *entre corchetes dobles*) en relación con el borrador 3 de SCC HEVC).

Tabla 5

ES 2 751 576 T3

	Descriptor
palette_coding(x0, y0, nCbS) {	
palettePredictionFinished = 0	
NumPredictedPaletteEntries = 0	
for(i = 0; i < PredictorPaletteSize && !palettePredictionFinished && NumPredictedPaletteEntries < palette_max_size; i++) {	
<b>palette_predictor_run</b>	ue(v)
if(palette_predictor_run != 1) {	
if(palette_predictor_run > 1)	
i += palette_predictor_run - 1	
PalettePredictorEntryReuseFlag[i] = 1	
NumPredictedPaletteEntries++	
} else	
palettePredictionFinished = 1	
}	
if(NumPredictedPaletteEntries < palette_max_size)	
<b>num_signalled_palette_entries</b>	ue(v)
numComps = (ChromaArrayType == 0) ? 1 : 3	
for(cldx = 0; cldx < numComps; cldx++)	
for(i = 0; i < num_signalled_palette_entries; i++)	
<b>palette_entry</b>	ae(v)
if(CurrentPaletteSize != 0)	
<b>palette_escape_val_present_flag</b>	ae(v)
[[if(palette_escape_val_present_flag) {}]]	
[[if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {}]]	
[[cu_qp_delta_palette_abs]]	[[ae(v)]]
[[if(cu_qp_delta_palette_abs)]]	
[[cu_qp_delta_palette_sign_flag]]	[[ae(v)]]
[[{}]]	
[[if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {}]]	
[[cu_chroma_qp_palette_offset_flag]]	[[ae(v)]]
[[if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)]]	
[[cu_chroma_qp_palette_offset_idx]]	[[ae(v)]]
[[{}]]	
[[{}]]	
if(MaxPaletteIndex > 0) {	
<b>num_palette_indices_idc</b>	ae(v)
for(i=0; i < NumPaletteIndices; i++) {	
<b>palette_index_idc</b>	ae(v)
PaletteIndexIdc[i] = palette_index_idc	
}	
<b>last_palette_run_type_flag</b>	ae(v)
<b>palette_transpose_flag</b>	ae(v)
}	
CurrNumIndices = 0	

ES 2 751 576 T3

	Descriptor
palette_coding(x0, y0, nCbS) {	
PaletteScanPos = 0	
while(PaletteScanPos < nCbS * nCbS) {	
xC = x0 + travScan[PaletteScanPos][0]	
yC = y0 + travScan[PaletteScanPos][1]	
if(PaletteScanPos > 0) {	
xcPrev = x0 + travScan[PaletteScanPos - 1][0]	
ycPrev = y0 + travScan[PaletteScanPos - 1][1]	
}	
PaletteRun = nCbS * nCbS - PaletteScanPos - 1	
if(MaxPaletteIndex > 0 && CurrNumIndices < NumPaletteIndices) {	
if(PaletteScanPos >= nCbS && palette_run_type_flag[xcPrev][ycPrev] != COPY_ABOVE_MODE && PaletteScanPos < nCbS * nCbS - 1) {	
<b>palette_run_type_flag[xC][yC]</b>	ae(v)
}	
readIndex = 0	
if(palette_run_type_flag[xC][yC] == COPY_INDEX_MODE && AdjustedMaxPaletteIndex > 0)	
readIndex = 1	
maxPaletteRun = nCbS * nCbS - PaletteScanPos - 1	
if(AdjustedMaxPaletteIndex > 0 && CurrNumIndices + readIndex < NumPaletteIndices    palette_run_type_flag[xC][yC] != last_palette_run_type_flag))	
if(maxPaletteRun > 0) {	
<b>palette_run_msb_id_plus1</b>	ae(v)
if(palette_run_msb_id_plus1 > 1)	
<b>palette_run_refinement_bits</b>	ae(v)
}	
CurrNumIndices += readIndex	
}	
runPos = 0	
while (runPos <= paletteRun) {	
xR = x0 + travScan[PaletteScanPos][0]	
yR = y0 + travScan[PaletteScanPos][1]	
if(palette_run_type_flag[xC][yC] == COPY_INDEX_MODE) {	
PaletteSampleMode[xR][yR] = COPY_INDEX_MODE	
PaletteIndexMap[xR][yR] = CurrPaletteIndex	
} else {	
PaletteSampleMode[xR][yR] = COPY_ABOVE_MODE	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xR][yR - 1]	
}	
runPos++	
PaletteScanPos++	
}	

	Descriptor
palette_coding(x0, y0, nCbS) {	
}	
if(palette_escape_val_present_flag) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
<b>cu_qp_delta_palette_abs</b>	ae(v)
if(cu_qp_delta_palette_abs)	
<b>cu_qp_delta_palette_sign_flag</b>	ae(v)
}	
if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetcoded) {	
<b>cu_chroma_qp_palette_offset_flag</b>	ae(v)
if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
<b>cu_chroma_qp_palette_offset_idx</b>	ae(v)
}	
sPos = 0	
while( sPos < nCbS * nCbS) {	
xC = x0 + travScan[sPos][0]	
yC = y0 + travScan[sPos][1]	
if(PaletteIndexMap[xC][yC] == MaxPaletteIndex) {	
for(cldx = 0; cldx < numComps; cldx++)	
if(cldx == 0    (xR % 2 == 0 && yR%2 == 0 && ChromaArrayType == 1)    (xR % 2 == 0 && ChromaArrayType == 2)    ChromaArrayType == 3) {	
<b>palette_escape_val</b>	ae(v)
PaletteEscapeVal[cldx][xC][yC] = palette_escape_val	
}	
}	
sPos++	
}	
}	

5 **[0034]** Las técnicas para codificación basada en paleta de datos de vídeo se pueden usar con una o más técnicas de codificación diferentes, tales como técnicas para codificación interpredictiva o intrapredictiva. Por ejemplo, como se describe con más detalle a continuación, un codificador o descodificador, o un codificador-descodificador combinado (códec), puede estar configurado para realizar una codificación interpredictiva e intrapredictiva, así como una codificación basada en paleta.

10 **[0035]** En algunos ejemplos, las técnicas de codificación basada en paleta pueden estar configuradas para su uso con uno o más estándares de codificación de vídeo. Algunos ejemplos de estándares de codificación de vídeo incluyen, pero no se limitan a, ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluyendo sus extensiones de codificación de vídeo escalable (SVC) y de codificación de vídeo multivista (MVC).

15 **[0036]** Recientemente, el Equipo Mixto de Colaboración sobre Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC ha terminado el diseño de un nuevo estándar de codificación de vídeo, a saber la codificación de vídeo de alta eficacia (HEVC). Una copia del estándar HEVC terminado (es decir, ITU-T H.265, serie H: "AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services - Coding of moving video", abril de 2015) está disponible en <https://www.itu.int/rec/T-REC-H.265-201504-l/en>, (el "estándar HEVC" más adelante en el presente documento).

20

**[0037]** El JCT-VC también está elaborando una extensión de rango para HEVC, a saber, la codificación de contenido de pantalla HEVC (SCC). Un borrador reciente de SCC HEVC (Joshi *et al.*, "High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 4", Equipo Mixto de Colaboración sobre Codificación de Vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 21.ª reunión: Varsovia, PL, 19 de junio - 16 de junio de 2015, está disponible en [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/documents/21\\_Warsaw/wg11/JCTVC-U1005-v2.zip](http://phenix.it-sudparis.eu/jct/doc_end_user/documents/21_Warsaw/wg11/JCTVC-U1005-v2.zip), ("borrador 4 de SCC HEVC" más adelante en el presente documento).

**[0038]** Con respecto al marco de la HEVC, como ejemplo, las técnicas de codificación basada en paleta pueden estar configuradas para su uso como un modo de unidad de codificación (CU). En otros ejemplos, las técnicas de codificación basada en paleta pueden estar configuradas para su uso como un modo de unidad de predicción (PU) en el marco de la HEVC. En consecuencia, todos los procesos siguientes divulgados, descritos en el contexto de un modo CU, se pueden aplicar, de forma adicional o alternativa, a una PU. Sin embargo, estos ejemplos basados en HEVC no se deberían considerar como una restricción o limitación de las técnicas de codificación basada en paleta, descritas en el presente documento, ya que dichas técnicas se pueden aplicar para que funcionen independientemente o como parte de otros sistemas/estándares existentes o todavía por elaborar. En estos casos, la unidad para codificación de paleta pueden ser unos bloques cuadrados, unos bloques rectangulares o incluso unas regiones de forma no rectangular.

**[0039]** La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación de vídeo 10 que puede utilizar las técnicas de esta divulgación. Como se usa en el presente documento, el término "codificador de vídeo" se refiere genéricamente tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" se pueden referir genéricamente a codificación de vídeo o a descodificación de vídeo. El codificador de vídeo 20 y el descodificador de vídeo 30 del sistema de codificación de vídeo 10 representan ejemplos de dispositivos que pueden estar configurados para realizar técnicas para codificación de vídeo basada en paleta de acuerdo con diversos ejemplos descritos en esta divulgación. Por ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar configurados para codificar selectivamente diversos bloques de datos de vídeo, tales como unas CU o PU en codificación HEVC, usando codificación basada en paleta o bien codificación no basada en paleta. Los modos de codificación no basada en paleta se pueden referir a diversos modos de codificación temporal interpredictiva o modos de codificación espacial intrapredictiva, tales como los diversos modos de codificación especificados por el estándar HEVC.

**[0040]** Como se muestra en la FIG. 1, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 y un dispositivo de destino 14. El dispositivo de origen 12 genera datos de vídeo codificados. En consecuencia, el dispositivo de origen 12 se puede denominar dispositivo de codificación de vídeo o aparato de codificación de vídeo. El dispositivo de destino 14 puede descodificar los datos de vídeo codificados generados por el dispositivo de origen 12. En consecuencia, el dispositivo de destino 14 se puede denominar dispositivo de descodificación de vídeo o aparato de descodificación de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 pueden ser ejemplos de dispositivos de codificación de vídeo o aparatos de codificación de vídeo.

**[0041]** El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia gama de dispositivos, que incluye ordenadores de escritorio, dispositivos informáticos móviles, ordenadores portátiles, ordenadores de tableta, descodificadores multimedia, aparatos telefónicos tales como los denominados teléfonos "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores para vehículos o similares.

**[0042]** El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 por medio de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicación que permiten al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de origen 12 puede modular los datos de vídeo codificados de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. El uno o más medios de comunicación pueden incluir medios de comunicación inalámbrica y/o alámbrica, tales como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El uno o más medios de comunicación pueden formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global (por ejemplo, Internet). El uno o más medios de comunicación pueden incluir encaminadores, conmutadores, estaciones base u otros equipos que facilitan la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

**[0043]** En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacena datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento por medio de acceso de disco o de acceso de tarjeta. El medio de almacenamiento puede incluir una variedad de medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria *flash* u otros medios de almacenamiento digital adecuados para almacenar datos de vídeo codificados.

**[0044]** En otro ejemplo, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacena los datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder a datos de vídeo codificados almacenados en el servidor de archivos o en otro dispositivo de almacenamiento intermedio mediante transmisión continua o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir los datos de vídeo codificados a un dispositivo de destino 14. Entre los ejemplos de servidores de archivos se incluyen servidores web (por ejemplo, para una página web), servidores de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados en red (NAS) y unidades de disco local.

**[0045]** El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de una conexión de datos estándar, tal como una conexión a Internet. Los ejemplos de tipos de conexiones de datos pueden incluir canales inalámbricos (por ejemplo, conexiones wifi), conexiones alámbricas (por ejemplo, DSL, módem de cable, etc.) o combinaciones de ambas que sean adecuadas para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

**[0046]** Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como soporte de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, por medio de Internet, codificación de datos de vídeo para su almacenamiento en un medio de almacenamiento de datos, descodificación de datos de vídeo almacenados en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 puede estar configurado para admitir transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

**[0047]** La FIG. 1 es simplemente un ejemplo y las técnicas de esta divulgación se pueden aplicar 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 desde una memoria local, se transmiten en continuo por una red o similares. Un dispositivo de codificación de vídeo puede codificar y almacenar datos en una memoria, y/o un dispositivo de descodificación de vídeo puede recuperar y descodificar datos de una memoria. En muchos ejemplos, la codificación y la descodificación se realizan mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en una memoria y/o recuperan y descodifican datos de una memoria. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluidos ordenadores de escritorio, ordenadores portátiles, ordenadores de tableta, descodificadores multimedia, aparatos telefónicos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

**[0048]** El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar por medio de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente a un dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación, inalámbrica o alámbrica, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar una comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

**[0049]** De forma alternativa, los datos codificados se pueden facilitar desde la interfaz de salida 22 a un dispositivo de almacenamiento 19. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 19 mediante una interfaz de entrada. El dispositivo de almacenamiento 19 puede incluir cualquiera de una variedad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco duro, unos discos Blu-ray, unos DVD, unos CD-ROM, una memoria *flash*, una memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 19 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que puede contener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a unos datos de vídeo almacenados en el dispositivo de almacenamiento 19 por medio de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los ejemplos servidores de archivos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14



puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión wifi), una conexión alámbrica (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 dispositivo de almacenamiento 19 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

**[0050]** Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como apoyo a cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, por medio de Internet, codificación de vídeo digital para almacenamiento en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para admitir una transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

**[0051]** En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captación de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo captado previamente, una interfaz de entrada de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como el vídeo de origen, o una combinación de dichas fuentes. Como ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar uno de los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o alámbricas.

**[0052]** El codificador de vídeo 20 puede codificar el vídeo captado, precaptado o generado por ordenador. Los datos de vídeo codificados se pueden transmitir directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados también (o de forma alternativa) se pueden almacenar en un dispositivo de almacenamiento 19 para un posterior acceso por el dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción.

**[0053]** El dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados a través de un enlace 16. Los datos de vídeo codificados comunicados a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 19, pueden incluir una variedad de elementos sintácticos generados por el codificador de vídeo 20 para su uso por un descodificador de vídeo, tal como el descodificador de vídeo 30, al descodificar los datos de vídeo. Dichos elementos sintácticos se pueden incluir con los datos de vídeo codificados, transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

**[0054]** El dispositivo de visualización 32 puede estar integrado en, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 visualiza los datos de vídeo descodificados ante un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

**[0055]** El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con un estándar de compresión de vídeo, tal como el estándar HEVC recientemente terminado (y varias extensiones del mismo actualmente en desarrollo). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otros estándares privados o industriales, tales como el estándar ITU-T H.264, denominado de forma alternativa MPEG-4, parte 10, codificación avanzada de vídeo (AVC), o extensiones de dichos estándares. Sin embargo, las técnicas de esta divulgación no están limitadas a ningún estándar de codificación particular. Otros ejemplos de estándares de compresión de vídeo incluyen VP8 y VP9.

**[0056]** Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrados cada uno con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, o hardware y programas informáticos diferentes, para gestionar la codificación tanto de audio como de vídeo en una corriente de datos común o en corrientes de datos independientes. Cuando proceda, en algunos ejemplos, las unidades MUX-DEMUX pueden cumplir con el protocolo de multiplexador ITU H.223 u otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

5 **[0057]** El codificador de vídeo 20 y el descodificador de vídeo 30 se pueden implementar cada uno como cualquiera de una variedad de circuitos de codificador adecuados, tales como uno o más circuitos integrados que incluyen microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, programas informáticos, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en un programa informático, un dispositivo puede almacenar instrucciones para el programa informático en un medio no transitorio legible por ordenador adecuado, y ejecutar las instrucciones en hardware, tal como unos circuitos integrados, usando uno o más procesadores para realizar las técnicas de esta divulgación. El codificador de vídeo 20 y el descodificador de vídeo 10 30 pueden estar incluidos cada uno en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado formando parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

15 **[0058]** Como se ha indicado anteriormente, el equipo JCT-VC ha terminado recientemente de elaborar el estándar HEVC. Los trabajos de estandarización HEVC se basaron en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo con respecto a los dispositivos existentes, de acuerdo, por ejemplo, con ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación de intrapredicción, el HM puede proporcionar hasta treinta y cinco modos de codificación de intrapredicción.

20 **[0059]** En HEVC y otras especificaciones de codificación de vídeo, una secuencia de vídeo incluye típicamente una serie de imágenes. Las imágenes también se pueden denominar "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 se pueden denominar en el presente documento muestras de "croma". En otros casos, una imagen puede ser monocromática y puede incluir solo una matriz de muestras de luma.

25 **[0060]** 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 comprender un bloque de árbol de codificación de muestras de luma, dos correspondientes bloques de árbol de codificación de muestras de croma y unas estructuras sintácticas usadas para codificar las muestras de los bloques de árbol de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CTU puede comprender un solo bloque de árbol de codificación y unas estructuras sintácticas usadas para codificar las muestras del bloque de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque  $N \times N$  de muestras. Una CTU también se puede denominar "bloque de árbol" o LCU. Las CTU de HEVC pueden ser análogas en términos generales a los macrobloques de otros estándares, tales como H.264/AVC. Sin embargo, una CTU no está necesariamente limitada a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un segmento puede incluir un número 30 35 entero de CTU ordenadas consecutivamente en un orden de exploración de barrido.

40 **[0061]** Para generar una CTU codificada, el codificador de vídeo 20 puede realizar de forma recurrente una división de árbol cuaternario en los bloques de árbol de codificación de una CTU para dividir los bloques de árbol de codificación en bloques de codificación, de ahí el nombre "unidades de árbol de codificación". Un bloque de codificación puede ser un bloque  $N \times N$  de muestras. Una CU puede comprender 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. En imágenes monocromáticas o imágenes que tienen tres 45 50 planos de color separados, una CU puede comprender un único bloque de codificación y estructuras sintácticas usadas para codificar las muestras del bloque de codificación.

55 **[0062]** El codificador de vídeo 20 puede dividir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción es 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 comprender un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma y estructuras sintácticas usadas para predecir los bloques de predicción. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una PU puede comprender un único bloque de predicción y estructuras sintácticas usadas para predecir el bloque 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.

60 **[0063]** El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU. Si el codificador de vídeo 20 usa intrapredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a unas muestras descodificadas de la imagen asociada a la PU. Si el codificador de vídeo 20 usa interpredicción para generar los bloques predictivos de una PU, el codificador de vídeo 20 puede generar los bloques predictivos de la PU en base a unas muestras descodificadas de una o más imágenes distintas a la imagen asociada a la PU.

65 **[0064]** Después de que el codificador de vídeo 20 genere bloques predictivos de luma, Cb y Cr para una o más PU de una CU, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra del bloque residual de luma de la CU indica una diferencia entre una muestra de luma de uno de los bloques predictivos de luma

de la CU y una muestra correspondiente del 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 del bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb de uno de los bloques predictivos de Cb de la CU y una muestra correspondiente del 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 del bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr de uno de los bloques predictivos de Cr de la CU y una muestra correspondiente del bloque de codificación de Cr original de la CU.

**[0065]** Además, el codificador de vídeo 20 puede usar una división de árbol cuaternario para descomponer los bloques residuales de luma, Cb y Cr de una CU en uno o más bloques de transformada de luma, Cb y Cr. Un bloque de transformada es un bloque rectangular (por ejemplo, cuadrado o no cuadrado) de muestras a las que se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede comprender un bloque de transformada de muestras de luma, dos correspondientes bloques de transformada de muestras de croma y estructuras sintácticas usadas para transformar las muestras de bloques de transformada. Por tanto, cada TU de una CU puede estar asociada a un bloque de transformada de luma, un bloque de transformada de Cb y un bloque de transformada de Cr. El bloque de transformada de luma asociado a la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformada de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un subbloque del bloque residual de Cr de la CU. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, una TU puede comprender un único bloque de transformada y estructuras sintácticas usadas para transformar las muestras del bloque de transformada.

**[0066]** El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de luma de una TU para generar un bloque de coeficientes de luma para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cb de una TU para generar un bloque de coeficientes de Cb para la TU. El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de Cr de una TU para generar un bloque de coeficientes de Cr para la TU.

**[0067]** Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luma, un bloque de coeficientes de Cb o un bloque de coeficientes de Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación se refiere, en general, a un proceso en el que unos coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede realizar una codificación de entropía de unos elementos sintácticos que indican los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar una codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos sintácticos que indican los coeficientes de transformada cuantificados.

**[0068]** El codificador de vídeo 20 puede facilitar una corriente de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. La corriente de bits puede comprender una secuencia de unidades NAL. Una unidad NAL es una estructura sintáctica que contiene una indicación del tipo de datos de la unidad NAL y unos bytes que contienen esos datos en forma de una RBSP entremezclada como sea necesario con bits de prevención de emulación. Cada una de las unidades NAL incluye una cabecera de la unidad NAL y encapsula una RBSP. La cabecera de unidad NAL puede incluir un elemento sintáctico que indica un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por la cabecera de unidad NAL de una unidad NAL indica el tipo de la unidad NAL. Una RBSP puede ser una estructura sintáctica que contiene un número entero de bytes que están encapsulados dentro de una unidad NAL. En algunos casos, una RBSP incluye bits cero.

**[0069]** 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 PPS, un segundo tipo de unidad NAL puede encapsular una RBSP para un segmento codificado, un tercer tipo de unidad NAL puede encapsular una RBSP para mensajes SEI, y así sucesivamente. Las unidades NAL que encapsulan una RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes SEI) se pueden denominar unidades VCL NAL.

**[0070]** El descodificador de vídeo 30 puede recibir una corriente de bits generada por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede analizar la corriente de bits para obtener elementos sintácticos a partir de la corriente de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo en base a, al menos en parte, los elementos sintácticos obtenidos a partir de la corriente de bits. El proceso para reconstruir los datos de vídeo puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20. Además, el descodificador de vídeo 30 puede realizar una cuantificación inversa de bloques de coeficientes asociados a unas TU de una CU actual. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes para reconstruir bloques de transformada asociados a las TU de la CU actual. El descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para unas PU de la CU actual a unas muestras correspondientes de los bloques de transformada de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

- 5 **[0071]** En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar configurados para realizar una codificación basada en paleta. Por ejemplo, en la codificación basada en paleta, en lugar de realizar las técnicas de codificación intrapredictiva o interpredictiva descritas anteriormente, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar una de las denominadas paletas como una tabla de valores de color para representar los datos de vídeo del área particular (por ejemplo, un bloque dado). Cada píxel puede estar asociado a una entrada de la paleta que representa el color del píxel, por ejemplo, a un valor de luma (Y) y unos valores de croma (Cb y Cr). Por ejemplo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar un índice que relaciona el valor de píxel con el valor adecuado de la paleta.
- 10 **[0072]** En el ejemplo anterior, 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 para representar el valor de cada píxel y codificando la paleta con valores de índice para los píxeles relacionando el valor de píxel con la paleta. El descodificador de vídeo 30 puede obtener, a partir de una corriente de bits codificada, una paleta para un bloque, así como valores de índice para los píxeles del bloque. El descodificador de vídeo 30 puede relacionar los valores de índice de los píxeles con unas entradas de paleta para reconstruir los valores de píxel del bloque.
- 15 **[0073]** Unos aspectos de esta divulgación están dirigidos a la obtención de paletas, que se puede producir en el codificador y en el descodificador. Como ejemplo, el codificador de vídeo 20 puede obtener una paleta para un bloque actual obteniendo un histograma de los píxeles del bloque actual. En algunos ejemplos, el histograma se puede expresar como  $H = \{(v_i, f_i), i = \{0, 1, 2, \dots, M\}\}$  donde  $M+1$  es el número de valores de píxel diferentes en el bloque actual,  $v_i$  es un valor de píxel, y  $f_i$  es el número de apariciones de  $v_i$  (es decir, cuántos píxeles del bloque actual tienen un valor de píxel  $v_i$ ). En dichos ejemplos, el histograma representa en general un número de veces en el que aparece un valor de píxel en el bloque actual.
- 20 **[0074]** El codificador de vídeo 20 puede inicializar una o más variables al obtener el histograma. Como ejemplo, el codificador de vídeo 20 puede inicializar un índice de paleta  $idx$  a 0 (es decir, establecer  $idx = 0$ ). Como otro ejemplo, el codificador de vídeo 20 puede inicializar la paleta  $P$  para que esté vacía (es decir,  $P = \emptyset$ , estableciendo  $j = 0$ ).
- 25 **[0075]** El codificador de vídeo 20 puede ordenar el histograma, por ejemplo, en orden descendente, de modo que los píxeles que tienen más apariciones estén colocados cerca de la parte frontal de una lista de valores. Por ejemplo, el codificador de vídeo 20 puede ordenar  $H$  de acuerdo con el orden descendente de  $f_i$ , y la lista ordenada se puede expresar como  $H_o = \{(u_i, f_i), i = \{0, 1, 2, \dots, M\}, f_i \geq f_{i+1}\}$ . En este ejemplo, la lista ordenada incluye los valores de píxel que aparecen con más frecuencia en la parte frontal (superior) de la lista y los valores de píxel que aparecen con menos frecuencia en la parte posterior (inferior) de la lista.
- 30 **[0076]** El codificador de vídeo 20 puede copiar una o más entradas del histograma en la paleta. Como ejemplo, el codificador de vídeo 20 puede insertar la entrada del histograma con la mayor frecuencia en la paleta. Por ejemplo, el codificador de vídeo 20 puede insertar  $(j, u_j)$  en la paleta  $P$  (es decir,  $P = P \cup \{(idx, u_j)\}$ ). En algunos ejemplos, después de insertar la entrada en la paleta, el codificador de vídeo 20 puede evaluar la entrada del histograma con la siguiente mayor frecuencia para su inserción en la paleta. Por ejemplo, el codificador de vídeo 20 puede establecer  $idx = idx + 1, j = j + 1$ .
- 35 **[0077]** El codificador de vídeo 20 puede determinar si la entrada con la siguiente mayor frecuencia (es decir,  $u_{j+1}$ ) está dentro de la vecindad de cualquier píxel (es decir,  $x$ ) de la paleta (es decir,  $Distancia(u_{j+1}, x) < Thresh$ ). Por ejemplo, el codificador de vídeo 20 puede determinar si la entrada está dentro de la vecindad de cualquier píxel de la paleta determinando si un valor de la entrada está dentro de una distancia umbral de un valor de cualquier píxel de la paleta. En algunos ejemplos, el codificador de vídeo 20 puede seleccionar de manera flexible la función de distancia. Como ejemplo, el codificador de vídeo 20 puede seleccionar la función de distancia como una suma de diferencias absolutas (SAD) o una suma de errores cuadráticos de predicción (SSE) de los tres componentes de color (por ejemplo, cada uno de luminancia, crominancia de tono azul y crominancia de tono rojo), o un componente de color (por ejemplo, uno de luminancia, crominancia de tono azul o crominancia de tono rojo). En algunos ejemplos, el codificador de vídeo 20 puede seleccionar de forma flexible el valor umbral  $Thresh$ . Como ejemplo, el codificador de vídeo 20 puede seleccionar el valor umbral para que sea dependiente del parámetro de cuantificación (QP) del bloque actual. Como otro ejemplo, el codificador de vídeo 20 puede seleccionar el valor umbral para que sea dependiente del valor de  $idx$  o del valor de  $j$ .
- 40 **[0078]** Si el codificador de vídeo 20 determina que la entrada con la siguiente mayor frecuencia (es decir,  $u_{j+1}$ ) está dentro de la vecindad de cualquier píxel de la paleta, el codificador de vídeo 20 puede no insertar la entrada en el histograma. Si el codificador de vídeo 20 determina que la entrada con la siguiente mayor frecuencia (es decir,  $u_{j+1}$ ) no está dentro de la vecindad de ningún píxel de la paleta, el codificador de vídeo 20 puede insertar la entrada en el histograma.
- 45 **[0079]** El codificador de vídeo 20 puede continuar insertando entradas en la paleta hasta que se cumplan una o más condiciones. Algunos ejemplos de condiciones se dan cuando  $idx = M$ , cuando  $j = M$  o cuando el tamaño de la paleta es mayor que un valor predefinido.
- 50 **[0079]** El codificador de vídeo 20 puede continuar insertando entradas en la paleta hasta que se cumplan una o más condiciones. Algunos ejemplos de condiciones se dan cuando  $idx = M$ , cuando  $j = M$  o cuando el tamaño de la paleta es mayor que un valor predefinido.
- 55 **[0079]** El codificador de vídeo 20 puede continuar insertando entradas en la paleta hasta que se cumplan una o más condiciones. Algunos ejemplos de condiciones se dan cuando  $idx = M$ , cuando  $j = M$  o cuando el tamaño de la paleta es mayor que un valor predefinido.
- 60 **[0079]** El codificador de vídeo 20 puede continuar insertando entradas en la paleta hasta que se cumplan una o más condiciones. Algunos ejemplos de condiciones se dan cuando  $idx = M$ , cuando  $j = M$  o cuando el tamaño de la paleta es mayor que un valor predefinido.
- 65 **[0079]** El codificador de vídeo 20 puede continuar insertando entradas en la paleta hasta que se cumplan una o más condiciones. Algunos ejemplos de condiciones se dan cuando  $idx = M$ , cuando  $j = M$  o cuando el tamaño de la paleta es mayor que un valor predefinido.

**[0080]** La codificación basada en paleta puede tener una determinada cantidad de sobrecarga de señalización. Por ejemplo, se puede necesitar un número bits para señalar características de una paleta, tales como un tamaño de la paleta, así como la propia paleta. Además, se puede necesitar un número de bits para señalar valores de índice para los píxeles del bloque. Las técnicas de esta divulgación pueden, en algunos ejemplos, reducir el número de bits necesarios para señalar dicha información. Por ejemplo, las técnicas descritas en esta divulgación pueden incluir técnicas para diversas combinaciones de una o más de una señalización de modos de codificación basada en paleta, una transmisión de paletas, una predicción de paletas, una obtención de paletas y una transmisión de mapas de codificación basada en paleta y otros elementos sintácticos.

**[0081]** En algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden predecir una paleta usando otra paleta. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar una primera paleta que tiene unas primeras entradas que indican unos primeros valores de píxel. El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden a continuación determinar, en base a las primeras entradas de la primera paleta, una o más segundas entradas que indican unos 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.

**[0082]** Cuando se determinan las entradas de la segunda paleta en base a las entradas de la primera paleta, el codificador de vídeo 20 puede codificar una variedad de elementos sintácticos, que el descodificador de vídeo puede usar para reconstruir la segunda paleta. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos de una corriente de bits para indicar que una paleta completa (o unas 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 separada) se copia desde uno o más bloques vecinos del bloque que se está codificando actualmente. La paleta a partir de la cual se predicen (por ejemplo, se copian) entradas de la paleta actual del bloque actual se puede denominar paleta predictiva. La paleta predictiva puede contener entradas de paleta de uno o más bloques vecinos, que incluyen bloques vecinos en el espacio y/o bloques vecinos en un orden de exploración particular de los bloques. Por ejemplo, los bloques vecinos pueden estar localizados espacialmente a la izquierda (bloque vecino izquierdo) o encima (bloque vecino 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 muestra más frecuentes en un vecino causal del bloque actual. En otro ejemplo, los bloques vecinos pueden ser vecinos del bloque que se está codificando actualmente, de acuerdo con un orden de exploración en particular usado para codificar los bloques. Es decir, los bloques vecinos 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 vecinos desde los cuales se copia(n) la(s) paleta(s).

**[0083]** En algunos ejemplos, la predicción de paletas se puede realizar con entradas. 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 está incluida en la paleta para el bloque actual. Si el codificador de vídeo 20 no predice una entrada de la paleta 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.

**[0084]** Los elementos sintácticos descritos anteriormente se pueden denominar vectores de predicción de paletas. Por ejemplo, como se indica anteriormente, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden predecir una paleta para un bloque actual en base una o más paletas de bloques vecinos (denominados colectivamente paleta de referencia). Al generar la paleta de referencia, se puede usar la tecnología primero en entrar, primero en salir (FIFO) añadiendo la última paleta al frente de la cola. Si la cola sobrepasa un umbral predefinido, los elementos más antiguos se pueden separar de ella. Después de empujar nuevos elementos al frente de la cola, se puede aplicar un proceso de poda para eliminar elementos duplicados, contando desde el principio de la cola. Específicamente, en algunos ejemplos, el codificador de vídeo 20 puede codificar (y el descodificador de vídeo 30 puede descodificar) un vector 0-1 para indicar si los valores de píxel de la paleta de referencia se reúsan para la paleta actual. Como ejemplo, como se muestra en el ejemplo de la Tabla 6, una paleta de referencia puede incluir seis elementos (por ejemplo, seis valores de índice y unos valores de píxel respectivos).

Tabla 6

Índice	Valor de píxel
0	$v_0$
1	$v_1$
2	$v_2$
3	$v_3$
4	$v_4$

Índice	Valor de píxel
5	v <sub>5</sub>

5 En un ejemplo con propósitos ilustrativos, el codificador de vídeo 20 puede señalar un vector (1, 0, 1, 1, 1, 1) que indica que v<sub>0</sub>, v<sub>2</sub>, v<sub>3</sub>, v<sub>4</sub> y v<sub>5</sub> se reusan en la paleta actual, mientras que v<sub>1</sub> no se reusa. Además de reusar v<sub>0</sub>, v<sub>2</sub>, v<sub>3</sub>, v<sub>4</sub> y v<sub>5</sub>, el codificador de vídeo 20 puede añadir dos nuevos elementos a la paleta actual con índices 5 y 6. La paleta actual para este ejemplo se muestra a continuación en la Tabla 7.

Tabla 7

Marca de pred.	Índice	Valor de píxel
1	0	v <sub>0</sub>
0		
1	1	v <sub>2</sub>
1	2	v <sub>3</sub>
1	3	v <sub>4</sub>
1	4	v <sub>5</sub>
	5	u <sub>0</sub>
	6	u <sub>1</sub>

10 **[0085]** Para codificar el vector 0-1 de predicción de paleta, para cada elemento del vector, el codificador de vídeo 20 puede codificar un bit para representar su valor. Adicionalmente, el número de elementos de paleta que no se pueden predecir (por ejemplo, el número de nuevas entradas de paleta (u<sub>0</sub> y u<sub>1</sub> en el ejemplo de la Tabla 7 anterior)) se puede binarizar y señalar.

15 **[0086]** Otros aspectos de esta divulgación se refieren a la construcción y/o transmisión de un mapa que permite que el codificador de vídeo 20 y/o el descodificador de vídeo 30 determinen valores de píxel. Por ejemplo, otros aspectos de esta divulgación se refieren a la construcción y/o la transmisión de un mapa de índices que relacionan un píxel particular con una entrada de una paleta.

20 **[0087]** 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 propósitos ilustrativos, se va a suponer 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 una marca para cada posición de píxel de un bloque. El codificador de vídeo 20 puede establecer la marca en uno para la entrada (i, j), para indicar que el valor de píxel en la localización (i, j) es uno de los  
 25 valores de la paleta. Cuando se incluye un color en la paleta (es decir, la marca 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 el color en la paleta. Cuando el color del píxel no está incluido en la paleta (es decir, la marca es igual a cero), el codificador de vídeo 20 también puede codificar datos que indican un valor de muestra para el píxel, que se puede denominar píxel de escape. El descodificador de vídeo 30 puede obtener los datos descritos anteriormente a partir de una corriente de bits  
 30 codificada y usar los datos para determinar un índice de paleta y/o valor de píxel para una localización particular en un bloque.

**[0088]** En algunos casos, puede haber una correlación entre el índice de paleta al que se mapea un píxel en una posición dada y la probabilidad de que un píxel vecino se mapee al mismo índice de paleta. Es decir, cuando un píxel se mapea a un índice de paleta particular, la probabilidad de que uno o más píxeles vecinos (en términos de localización espacial) se mapeen al mismo índice de paleta puede ser relativamente alta.

**[0089]** En algunos ejemplos, 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, en relación con 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 estar configurados para determinar un primer valor de índice asociado a un primer píxel de 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 estar configurados para determinar, en base al primer valor de índice, uno o más segundos valores de índice asociados a uno o más segundos píxeles del bloque de datos de vídeo, y para  
 40 datos de vídeo. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden estar configurados para determinar, en base al primer valor de índice, uno o más segundos valores de índice asociados a uno o más segundos píxeles del bloque de datos de vídeo, y para  
 45 codificar el primer y el uno o más segundos píxeles del bloque de datos de vídeo. Por tanto, en este ejemplo, los índices de un mapa se pueden codificar en relación con uno o más índices diferentes del mapa.

5 [0090] Como se analiza anteriormente, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden usar varias técnicas diferentes para codificar valores de índice de un mapa en relación con otros índices del mapa. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden usar un modo de índice, copiar un modo superior y realizar una transición de modo para codificar valores de índice de un mapa en relación con otros índices del mapa.

10 [0091] En el "modo índice" de la codificación basada en paleta, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden señalar primero un índice de paleta. Si el índice es igual al tamaño de la paleta, esto indica que la muestra es una muestra de escape. En este caso, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden señalar el valor de la muestra o el valor de las muestras cuantificadas para cada componente. Por ejemplo, si el tamaño de la paleta es 4, para muestras no de escape, los índices de la paleta están en el intervalo [0, 3]. En este caso, un valor de índice de 4 puede significar una muestra de escape. Si el índice indica una muestra no de escape, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden indicar una longitud de recorrido, que puede especificar el número de muestras posteriores en orden de exploración que comparten el mismo índice, mediante un valor no negativo  $n-1$  que indica la longitud de recorrido, lo que significa que los siguientes  $n$  píxeles, incluyendo el actual, tienen el mismo índice de píxel que el primer índice señalado.

20 [0092] En el modo "copiar de superior" de la codificación basada en paleta, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden señalar un valor de longitud de recorrido no negativo  $m-1$  para indicar que para los siguientes  $m$  píxeles, incluyendo el píxel actual, los índices de paleta son los mismos que los de sus vecinos situados directamente encima, respectivamente. Debe tenerse en cuenta que el modo "copiar de superior" es diferente del modo "índice", en el sentido de que los índices de paleta podrían ser diferentes dentro del modo de recorrido "copiar de superior".

25 [0093] Como se analiza anteriormente, en algunos ejemplos, puede ser deseable agrupar períodos de derivación (es decir, para incrementar el rendimiento CABAC). De acuerdo con una o más técnicas de esta divulgación, el codificador de vídeo 20 puede codificar, y el descodificador de vídeo 30 puede descodificar, elementos sintácticos usados para definir una paleta actual de modo que los elementos sintácticos que se codifican usando el modo de derivación se agrupen juntos. Por ejemplo, en lugar de codificar uno o más elementos sintácticos relacionados con un parámetro de cuantificación (QP) delta y/o unos desplazamientos de QP de croma para un bloque actual de datos de vídeo antes de un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para el bloque actual de datos de vídeo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo después del elemento sintáctico que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de vídeo. De esta manera, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar un grupo más grande de elementos sintácticos usando el modo de derivación, lo que puede incrementar el rendimiento CABAC.

40 [0094] En algunos ejemplos, el uno o más elementos sintácticos relacionados con un QP delta para el bloque actual de datos de vídeo pueden incluir un elemento sintáctico que especifica el valor absoluto de una diferencia entre un QP de luma para el bloque actual de datos de vídeo y un predictor del QP de luma para el bloque actual (por ejemplo, **cu\_qp\_delta\_palette\_abs**) y un elemento sintáctico que especifica un signo de la diferencia entre el QP de luma para el bloque actual de datos de vídeo y el predictor del QP de luma para el bloque actual (por ejemplo, **cu\_qp\_delta\_palette\_sign\_flag**). En algunos ejemplos, el uno o más elementos sintácticos relacionados con desplazamientos de QP de croma para el bloque actual de datos de vídeo pueden incluir un elemento sintáctico que indica si unas entradas de una o más listas de desplazamiento se añaden al QP de luma para el bloque actual para determinar unos QP de croma para el bloque actual (por ejemplo, **cu\_chroma\_qp\_palette\_offset\_flag**), y un elemento sintáctico que especifica un índice de una entrada de cada una de la una o más listas de desplazamiento que se añaden al QP de luma para el bloque actual para determinar unos QP de croma para el bloque actual (por ejemplo, **cu\_chroma\_qp\_palette\_offset\_idx**). Así pues, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden estar configurados cada uno para codificar un elemento sintáctico **palette\_transpose\_flag** en una primera posición de una corriente de bits y codificar un elemento sintáctico **cu\_qp\_delta\_palette\_abs**, un elemento sintáctico **cu\_qp\_delta\_palette\_sign\_flag**, un elemento sintáctico **cu\_chroma\_qp\_palette\_offset\_flag**, y un elemento sintáctico **cu\_chroma\_qp\_palette\_offset\_idx** en una segunda posición de la corriente de bits que está después de la primera posición.

60 [0095] La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas de esta divulgación. La FIG. 2 se proporciona con propósitos explicativos y no se debería considerar limitativa de las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con propósitos explicativos, esta divulgación describe un codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros estándares o procedimientos de codificación.

65 [0096] El codificador de vídeo 20 representa un ejemplo de dispositivo que puede estar configurado para realizar técnicas para codificación de vídeo basada en paleta de acuerdo con diversos ejemplos descritos en esta divulgación. Por ejemplo, el codificador de vídeo 20 puede estar configurado para codificar selectivamente diversos bloques de datos de vídeo, tales como unas CU o PU en codificación HEVC, usando codificación basada en paleta o codificación

no basada en paleta. Los modos de codificación no basada en paleta se pueden referir a diversos modos de codificación temporal interpredictiva o modos de codificación espacial intrapredictiva, tales como los diversos modos de codificación especificados por el estándar HEVC. El codificador de vídeo 20, en un ejemplo, puede estar configurado 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 a unas entradas de la paleta correspondientes, respectivamente, a los valores de píxel seleccionados. El descodificador de vídeo 30 puede usar la información señalada para descodificar datos de vídeo.

**[0097]** En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de procesamiento de predicción 100, una unidad de generación residual 102, una unidad de procesamiento de transformada 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de procesamiento de transformada inversa 110, una unidad de reconstrucción 112, una unidad de filtro 114, una memoria intermedia de imágenes descodificadas 116 y una unidad de codificación de entropía 118. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de interpredicción 120 y una unidad de procesamiento de intrapredicción 126. La unidad de procesamiento de interpredicción 120 incluye una unidad de estimación de movimiento y una unidad de compensación de movimiento (no mostradas). El codificador de vídeo 20 también incluye una unidad de codificación basada en paleta 122, configurada para realizar diversos aspectos de las técnicas de codificación basada en paleta descritas en esta divulgación. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales.

**[0098]** El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede codificar cada CTU en un segmento de una imagen de los datos de vídeo. Cada una de las CTU puede estar asociada a unos bloques de árbol de codificación (CTB) de luma de igual tamaño y a unos CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede realizar una división de árbol cuaternario para dividir los CTB de la CTU en bloques progresivamente más pequeños. Los bloques más pequeños pueden ser bloques de codificación de unas CU. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un CTB asociado a una CTU en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, y así sucesivamente.

**[0099]** El codificador de vídeo 20 puede codificar las CU de una CTU para generar representaciones codificadas de las CU (es decir, CU codificadas). Como parte de la codificación de una CU, la unidad de procesamiento de predicción 100 puede dividir los bloques de codificación asociados a la CU entre una o más PU de la CU. Por tanto, cada PU puede estar asociada a un bloque de predicción de luma y a unos correspondientes bloques de predicción de croma. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir unas PU que tienen diversos tamaños. Como se ha indicado anteriormente, el tamaño de una CU se puede referir al tamaño del bloque de codificación de luma de la CU, y el tamaño de una PU se puede referir al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular sea de  $2N \times 2N$ , el codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir tamaños de PU de  $2N \times 2N$  o  $N \times N$  para intrapredicción, y tamaños de PU simétricas de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$  o similares para interpredicción. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden admitir una división asimétrica para tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$  para interpredicción.

**[0100]** La unidad de procesamiento de interpredicción 120 puede generar datos predictivos para una PU realizando una interpredicción en cada PU de una CU. Los datos predictivos para la PU pueden incluir bloques de muestras predictivos de la PU e información de movimiento para la PU. La unidad de procesamiento de interpredicción 120 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un segmento I, un segmento P o un segmento B. En un segmento I, todas las PU se intrapredicen. Por consiguiente, si la PU está en un segmento I, la unidad de procesamiento de interpredicción 120 no realiza interpredicción en la PU. Por tanto, para bloques codificados en el modo I, el bloque predicho se forma usando predicción espacial a partir de bloques vecinos previamente codificados dentro de la misma trama.

**[0101]** Si una PU está en un segmento P, la unidad de estimación de movimiento de la unidad de procesamiento de interpredicción 120 puede buscar en las imágenes de referencia de una lista de imágenes de referencia (por ejemplo, "RefPicListO") una región de referencia para la PU. La región de referencia para la PU puede ser una región, dentro de una imagen de referencia, que contiene bloques de muestras que se corresponden más estrechamente con los bloques de muestras de la PU. La unidad de estimación de movimiento puede generar un índice de referencia que indica una posición en RefPicListO de la imagen de referencia que contiene la región de referencia para la PU. Además, la unidad de estimación de movimiento puede generar un MV que indica un desplazamiento espacial entre un bloque de codificación de la PU y una localización de referencia asociada a la región de referencia. Por ejemplo, el MV puede ser un vector bidimensional que proporciona un desplazamiento desde las coordenadas de la imagen descodificada actual hasta las coordenadas de una imagen de referencia. La unidad de estimación de movimiento puede facilitar el índice de referencia y el MV como la información de movimiento de la PU. La unidad de compensación de movimiento de la unidad de procesamiento de interpredicción 120 puede generar los bloques de muestras predictivos de la PU en base a unas muestras reales o interpoladas en la localización de referencia indicada por el vector de movimiento de la PU.



[0102] Si una PU está en un segmento B, la unidad de estimación de movimiento puede realizar unipredicción o bipredicción para la PU. Para realizar la unipredicción para la PU, la unidad de estimación de movimiento puede buscar en las imágenes de referencia de RefPicListO o de una segunda lista de imágenes de referencia (“RefPicList1”) una región de referencia para la PU. La unidad de estimación de movimiento puede facilitar, como la información de movimiento de la PU, un índice de referencia que indica una posición en RefPicListO o en RefPicList1 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 a la región de referencia, y uno o más indicadores de dirección de predicción que indican si la imagen de referencia está en RefPicListO o en RefPicList1. La unidad de compensación de movimiento de la unidad de procesamiento de interpredicción 120 puede generar los bloques de muestras predictivos de la PU en base, al menos en parte, a muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.

[0103] Para realizar una interpredicción bidireccional para una PU, la unidad de estimación de movimiento puede buscar en las imágenes de referencia de RefPicListO una región de referencia para la PU y también puede buscar en las imágenes de referencia de RefPicList1 otra región de referencia para la PU. La unidad de estimación de movimiento puede generar índices de imágenes de referencia que indican posiciones en RefPicListO y en RefPicList1 de las imágenes de referencia que contienen las regiones de referencia. Además, la unidad de estimación de movimiento puede generar unos MV que indican desplazamientos espaciales entre la localización de referencia asociada a las regiones de referencia y un bloque de muestras de la PU. La información de movimiento de la PU puede incluir los índices de referencia y los MV de la PU. La unidad de compensación de movimiento puede generar los bloques de muestras predictivos de la PU en base, al menos en parte, a muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.

[0104] De acuerdo con diversos ejemplos de esta divulgación, el codificador de vídeo 20 puede estar configurado para realizar una codificación basada en paleta. Con respecto al marco de la HEVC, como ejemplo, las técnicas de codificación basada en paleta pueden estar configuradas para su uso como un modo de unidad de codificación (CU). En otros ejemplos, las técnicas de codificación basada en paleta pueden estar configuradas para su uso como un modo PU en el marco de la HEVC. En consecuencia, todos los procesos divulgados, descritos en el presente documento (a lo largo de esta divulgación) en el contexto de un modo de CU, se pueden aplicar, de forma adicional o alternativa, a las PU. Sin embargo, estos ejemplos basados en HEVC no se deberían considerar como una restricción o limitación de las técnicas de codificación basada en paleta, descritas en el presente documento, ya que dichas técnicas se pueden aplicar para que funcionen independientemente o como parte de otros sistemas/estándares existentes o todavía por elaborar. En estos casos, la unidad para codificación de paleta pueden ser unos bloques cuadrados, unos bloques rectangulares o incluso unas regiones de forma no rectangular.

[0105] La unidad de codificación basada en paleta 122, por ejemplo, puede realizar una codificación basada en paleta cuando se selecciona un modo de codificación basada en paleta, por ejemplo, para una CU o PU. Por ejemplo, la unidad de codificación basada en paleta 122 puede estar configurada para generar una paleta que tiene entradas que indican valores de píxel, seleccionar valores de píxel de 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 a entradas de la paleta correspondiente, respectivamente, a los valores de píxel seleccionados. Aunque se describe que la unidad de codificación basada en paleta 122 realiza diversas funciones, otras unidades de procesamiento, o una combinación de diferentes unidades de procesamiento, pueden realizar algunas o la totalidad de dichas funciones.

[0106] La unidad de codificación basada en paleta 122 puede generar elementos sintácticos para definir una paleta para un bloque de datos de vídeo. Algunos ejemplos de elementos sintácticos que la unidad de codificación basada en paleta 122 puede generar para definir una paleta actual para un bloque actual de datos de vídeo incluyen, pero no se limitan a, un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de la paleta actual (p. ej., **palette\_transpose\_flag**), uno o más elementos sintácticos relacionados con un parámetro de cuantificación (QP) delta (p. ej., **cu\_qp\_delta\_palette\_abs**, **cu\_qp\_delta\_palette\_sign\_flag**, **cu\_chroma\_qp\_palette\_offset\_flag**, y/o **cu\_chroma\_qp\_palette\_offset\_idx**), uno o más elementos sintácticos relacionados con unos desplazamientos de QP de croma para el bloque actual de datos de vídeo, uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si unas entradas de una paleta predictora se reúsan en la paleta actual (por ejemplo, **palette\_predictor\_run**), uno o más elementos sintácticos que indican un número de entradas de la paleta actual que se señalan explícitamente (por ejemplo, **num\_signalled\_palette\_entries**), uno o más elementos sintácticos que indican un valor de un componente de una entrada de paleta de la paleta actual (por ejemplo, **palette\_entry**), uno o más elementos sintácticos que indican si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape (por ejemplo, **palette\_escape\_val\_present\_flag**), uno o más elementos sintácticos que indican un número de entradas de la paleta actual que se señalan o deducen explícitamente (por ejemplo, **num\_palette\_indices\_idc**), y uno o más elementos sintácticos que indican índices de una matriz de entradas de paleta actual (por ejemplo, **palette\_index\_idc**). La unidad de codificación basada en paleta 122 puede facilitar los elementos sintácticos generados que definen la paleta actual para el bloque actual a uno o más componentes diferentes del codificador de vídeo 20, tal como la unidad de codificación de entropía 118.

- 5 **[0107]** Por consiguiente, el codificador de vídeo 20 puede estar configurado para codificar bloques de datos de vídeo usando modos de codificación basada en paleta como se describe en esta divulgación. El codificador de vídeo 20 puede codificar selectivamente un bloque de datos de vídeo usando un modo de codificación basada en paleta, o codificar un bloque de datos de vídeo usando un modo diferente, por ejemplo, tal como un modo de codificación HEVC interpredictiva o intrapredictiva. El bloque de datos de vídeo puede ser, por ejemplo, una CU o PU generada de acuerdo con un proceso de codificación HEVC. Un codificador de vídeo 20 puede codificar algunos bloques con modos de codificación de predicción temporal interpredictiva o espacial intrapredictiva y descodificar otros bloques con el modo de codificación basada en paleta.
- 10 **[0108]** La unidad de procesamiento de intrapredicción 126 puede generar datos predictivos para una PU realizando una intrapredicción en la PU. Los datos predictivos para la PU pueden incluir bloques de muestras predictivos para la PU y diversos elementos sintácticos. La unidad de procesamiento de intrapredicción 126 puede realizar una intrapredicción en las PU en segmentos I, segmentos P y segmentos B.
- 15 **[0109]** Para realizar una intrapredicción en una PU, la unidad de procesamiento de intrapredicción 126 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos predictivos para la PU. Para usar un modo de intrapredicción para generar un conjunto de datos predictivos para la PU, la unidad de procesamiento de intrapredicción 126 puede extender muestras de bloques de muestras de PU vecinas por los bloques de muestras de la PU en una dirección asociada al modo de intrapredicción. Las PU vecinas pueden estar encima, encima y a la derecha, encima y a la izquierda, o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha y de arriba abajo, para las PU, CU y CTU. La unidad de procesamiento de intrapredicción 126 puede usar diversos números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional. En algunos ejemplos, el número de modos de intrapredicción puede depender del tamaño de la región asociada a la PU.
- 20 **[0110]** La unidad de procesamiento de predicción 100 puede seleccionar los datos predictivos para las PU de una CU de entre los datos predictivos generados por la unidad de procesamiento de interpredicción 120 para las PU, o los datos predictivos generados por la unidad de procesamiento de intrapredicción 126 para las PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos predictivos para las PU de la CU en base a unas 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 **[0111]** La unidad de generación residual 102 puede generar, en base al bloque de codificación de luma, Cb y Cr de una CU y los bloques predictivos seleccionados de luma, Cb y Cr de las PU de la CU, unos bloques residuales de luma, Cb y Cr de la CU. Por ejemplo, la unidad de generación residual 102 puede generar los bloques residuales de la CU de modo que cada muestra de los bloques residuales tenga un valor igual a una diferencia entre una muestra de un bloque de codificación de la CU y una muestra correspondiente de un correspondiente bloque de muestras predictivo seleccionado de una PU de la CU.
- 30 **[0112]** La unidad de procesamiento de transformada 104 puede realizar una división de árbol cuaternario para dividir los bloques residuales asociados a una CU en bloques de transformada asociados a unas TU de la CU. Por tanto, una TU puede estar asociada a un bloque de transformada de luma y a dos bloques de transformada de croma. Los tamaños y las posiciones de los bloques de transformada de luma y croma de unas TU de una CU pueden o no estar basados en los tamaños y las posiciones de los bloques de predicción de las PU de la CU. Una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados a cada una de las regiones. Las TU de una CU pueden corresponder a nodos hoja del RQT.
- 35 **[0113]** La unidad de procesamiento de transformada 104 puede generar bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas a un bloque de transformada asociado a una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar al bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 104 no aplica transformadas a un bloque de transformada. En dichos ejemplos, el bloque de transformada se puede tratar como un bloque de coeficientes de transformada.
- 40 **[0114]** La unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de  $n$  bits puede redondearse a la baja hasta un coeficiente de transformada de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ . La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado a una TU de una CU en base a un valor de parámetro de cuantificación (QP) asociado a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados a una CU, ajustando el valor de QP asociado a la CU. La cuantificación puede introducir pérdida de información, por tanto unos coeficientes de transformada cuantificados pueden tener una precisión menor que los originales.
- 45 **[0113]** La unidad de procesamiento de transformada 104 puede generar bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas a un bloque de transformada asociado a una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar al bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 104 no aplica transformadas a un bloque de transformada. En dichos ejemplos, el bloque de transformada se puede tratar como un bloque de coeficientes de transformada.
- 50 **[0114]** La unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de  $n$  bits puede redondearse a la baja hasta un coeficiente de transformada de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ . La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado a una TU de una CU en base a un valor de parámetro de cuantificación (QP) asociado a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados a una CU, ajustando el valor de QP asociado a la CU. La cuantificación puede introducir pérdida de información, por tanto unos coeficientes de transformada cuantificados pueden tener una precisión menor que los originales.
- 55 **[0113]** La unidad de procesamiento de transformada 104 puede generar bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas a un bloque de transformada asociado a una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar al bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 104 no aplica transformadas a un bloque de transformada. En dichos ejemplos, el bloque de transformada se puede tratar como un bloque de coeficientes de transformada.
- 60 **[0114]** La unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de  $n$  bits puede redondearse a la baja hasta un coeficiente de transformada de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ . La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado a una TU de una CU en base a un valor de parámetro de cuantificación (QP) asociado a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados a una CU, ajustando el valor de QP asociado a la CU. La cuantificación puede introducir pérdida de información, por tanto unos coeficientes de transformada cuantificados pueden tener una precisión menor que los originales.
- 65 **[0114]** La unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de  $n$  bits puede redondearse a la baja hasta un coeficiente de transformada de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ . La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado a una TU de una CU en base a un valor de parámetro de cuantificación (QP) asociado a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados a una CU, ajustando el valor de QP asociado a la CU. La cuantificación puede introducir pérdida de información, por tanto unos coeficientes de transformada cuantificados pueden tener una precisión menor que los originales.

[0115] La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformada inversa 110 pueden aplicar una cuantificación inversa y transformadas inversas al bloque de coeficientes, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes. La unidad de reconstrucción 112 puede añadir el bloque residual reconstruido a unas muestras correspondientes de uno o más bloques de muestras predictivos generados por la unidad de procesamiento de predicción 100 para generar un bloque de transformada reconstruido asociado a una TU. Reconstruyendo bloques de transformada para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

[0116] La unidad de filtro 114 puede realizar una o más operaciones antibloqueo para reducir los artefactos de bloque en los bloques de codificación asociados a una CU. La memoria intermedia de imágenes descodificadas 116 puede almacenar los bloques de codificación reconstruidos después de que la unidad de filtro 114 realice la una o más operaciones antibloqueo en los bloques de codificación reconstruidos. La unidad de procesamiento de interpredicción 120 puede usar una imagen de referencia que contiene los bloques de codificación reconstruidos para realizar una interpredicción en las PU de otras imágenes. Además, la unidad de procesamiento de intrapredicción 126 puede usar bloques de codificación reconstruidos en la memoria intermedia de imágenes descodificadas 116 para realizar una intrapredicción en otras PU de la misma imagen que la CU.

[0117] La unidad de codificación de entropía 118 puede recibir datos desde otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación de entropía 118 puede recibir bloques de coeficientes desde la unidad de cuantificación 106 y puede recibir elementos sintácticos desde la unidad de procesamiento de predicción 100. La unidad de codificación de entropía 118 puede realizar una o más operaciones de codificación de entropía en los datos para generar datos sometidos a codificación de entropía. Por ejemplo, la unidad de codificación de entropía 118 puede realizar una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), una operación de codificación de entropía de división de intervalos de probabilidad (PIPE), una operación de codificación Golomb exponencial u otro tipo de operación de codificación de entropía en los datos. El codificador de vídeo 20 puede facilitar una corriente de bits que incluye datos sometidos a codificación de entropía, generados por la unidad de codificación de entropía 118. Por ejemplo, la corriente de bits puede incluir datos que representan un RQT para una CU.

[0118] Como se analiza anteriormente, la unidad de codificación basada en paleta 122 puede facilitar los elementos sintácticos generados que definen la paleta actual para el bloque actual a la unidad de codificación de entropía 118. La unidad de codificación de entropía 118 puede codificar uno o más períodos de los elementos sintácticos recibidos desde la unidad de codificación basada en paleta 122 usando CABAC con contextos y uno o más períodos de los elementos sintácticos recibidos desde la unidad de codificación basada en paleta 122 usando CABAC sin contextos (es decir, un modo de derivación). En algunos ejemplos, la unidad de codificación de entropía 118 puede codificar los períodos de los elementos sintácticos usando contextos o un modo de derivación como se define anteriormente en la Tabla 2.

[0119] Como se analiza anteriormente, puede ser deseable agrupar los períodos codificados en derivación para incrementar el rendimiento CABAC. En el borrador 3 de SCC, los períodos de los elementos sintácticos **palette\_predictor\_run**, **num\_signalled\_palette\_entries**, **palette\_entry** y **palette\_escape\_val\_present\_flag** se someten a codificación de derivación y se agrupan. Sin embargo, aunque los períodos de los elementos sintácticos **num\_palette\_indices\_idc**, y **palette\_index\_idc** también se someten a codificación de derivación, no se agrupan con los períodos de los elementos sintácticos **palette\_predictor\_run**, **num\_signalled\_palette\_entries**, **palette\_entry** y **palette\_escape\_val\_present\_flag**. En cambio, en el borrador 3 de SCC HEVC, los elementos sintácticos **num\_palette\_indices\_idc**, y **palette\_index\_idc** están separados de los elementos sintácticos **palette\_predictor\_run**, **num\_signalled\_palette\_entries**, **palette\_entry**, y **palette\_escape\_val\_present\_flag** por uno o más elementos sintácticos relacionados con un parámetro de cuantificación (QP) delta y/o unos desplazamientos de QP de croma para un bloque actual de datos de vídeo (es decir, **cu\_qp\_delta\_palette\_abs**, **cu\_qp\_delta\_palette\_sign\_flag**, **cu\_chroma\_qp\_palette\_offset\_flag** y **cu\_chroma\_qp\_palette\_offset\_idx**) y un elemento sintáctico que indica si un proceso de transposición se aplica a los índices de la paleta para el bloque actual de datos de vídeo (es decir, **palette\_transpose\_flag**).

[0120] De acuerdo con una o más técnicas de esta divulgación, la unidad de codificación de entropía 118 puede codificar los elementos sintácticos usados para definir una paleta actual de modo que unos elementos sintácticos que se codifican usando un modo de derivación se codifican consecutivamente. Por ejemplo, en lugar de separar los períodos de los elementos sintácticos **palette\_predictor\_run**, **num\_signalled\_palette\_entries**, **palette\_entry** y **palette\_escape\_val\_present\_flag** y los períodos de los elementos sintácticos **num\_palette\_indices\_idc** y **palette\_index\_idc**, la unidad de codificación de entropía 118 puede codificar uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo después de un elemento sintáctico que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de vídeo, de modo que los períodos de los elementos sintácticos **palette\_predictor\_run**, **num\_signalled\_palette\_entries**, **palette\_entry** y **palette\_escape\_val\_present\_flag**, **num\_indices\_idc** y **palette\_index\_idc** se agrupan. De esta manera, se puede incrementar el rendimiento CABAC de la unidad de codificación de entropía 118.

**[0121]** La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que está configurado para implementar las técnicas de esta divulgación. La FIG. 3 se proporciona con propósitos explicativos y no se limita a las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con propósitos explicativos, esta divulgación describe un descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otros estándares o procedimientos de codificación.

**[0122]** El codificador de vídeo 30 representa un ejemplo de dispositivo que puede estar configurado para realizar técnicas para codificación de vídeo basada en paleta, de acuerdo con diversos ejemplos descritos en esta divulgación. Por ejemplo, el descodificador de vídeo 30 puede estar configurado para descodificar selectivamente diversos bloques de datos de vídeo, tales como las CU o las PU en la codificación HEVC, usando codificación basada en paleta o bien codificación no basada en paleta. Los modos de codificación no basada en paleta se pueden referir a diversos modos de codificación temporal interpredictiva o modos de codificación espacial intrapredictiva, tales como los diversos modos de codificación especificados por el estándar HEVC. El descodificador de vídeo 30, en un ejemplo, puede estar configurado para generar una paleta que tiene entradas que indican valores de píxel, recibir información que asocia al menos algunas posiciones de un bloque de datos de vídeo a entradas de la paleta, seleccionar valores de píxel en la paleta en base a la información y reconstruir valores de píxel del bloque en base a los valores de píxel seleccionados.

**[0123]** En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación de entropía 150, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de procesamiento de transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 160 y una memoria intermedia de imágenes descodificadas 162. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 164 y una unidad de procesamiento de intrapredicción 166. El descodificador de vídeo 30 también incluye una unidad de descodificación basada en paleta 165 configurada para realizar diversos aspectos de las técnicas de codificación basada en paleta descritas en esta divulgación. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

**[0124]** En algunos ejemplos, el descodificador de vídeo 30 puede incluir además memoria de datos de vídeo 149. La memoria de datos de vídeo 149 puede almacenar datos de vídeo, tales como una corriente de bits de vídeo codificada que los componentes del descodificador de vídeo 30 van a descodificar. Los datos de vídeo almacenados en memoria de datos de vídeo 149 se pueden obtener, por ejemplo, a partir del canal 16, por ejemplo, desde una fuente de vídeo local, tal como una cámara, por medio de una comunicación alámbrica o inalámbrica de datos de vídeo, o accediendo a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 149 puede formar una memoria intermedia de imágenes codificadas (CPB) que almacena datos de vídeo codificados a partir de una corriente de bits de vídeo codificada. La CPB 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 por el descodificador de vídeo 30, por ejemplo, en modos de intracodificación o intercodificación. La memoria de datos de vídeo 149 puede estar formada por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El mismo dispositivo de memoria o unos dispositivos de memoria separados pueden proporcionar una memoria de datos de vídeo 149 y una memoria intermedia de imágenes descodificadas 162. En diversos ejemplos, la memoria de datos de vídeo 149 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera del chip con respecto a esos componentes.

**[0125]** Una memoria intermedia de imágenes codificadas (CPB) puede recibir y almacenar datos de vídeo codificados (por ejemplo, unidades NAL) de una corriente de bits. La unidad de descodificación de entropía 150 puede recibir datos de vídeo codificados (por ejemplo, unidades NAL) desde la CPB y analizar sintácticamente las unidades NAL para descodificar elementos sintácticos. La unidad de descodificación de entropía 150 puede realizar la descodificación de entropía de unos elementos sintácticos sometidos a codificación de entropía en las unidades NAL. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de procesamiento de transformada inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 160 pueden generar datos de vídeo descodificados en base a los elementos sintácticos extraídos de la corriente de bits.

**[0126]** Las unidades NAL de la corriente de bits pueden incluir unidades NAL de segmentos codificados. Como parte de la descodificación de la corriente de bits, la unidad de descodificación de entropía 150 puede extraer y realizar la descodificación de entropía unos elementos sintácticos de las unidades NAL de segmentos codificados. Cada uno de los segmentos codificados puede incluir una cabecera de segmento y datos de segmento. La cabecera de segmento puede contener elementos sintácticos pertenecientes a un segmento. Los elementos sintácticos de la cabecera de segmento pueden incluir un elemento sintáctico que identifica un PPS asociado a una imagen que contiene el segmento.

**[0127]** Además de descodificar los elementos sintácticos de la corriente de bits, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU.

Realizando la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir bloques residuales de la CU.

5 **[0128]** Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede realizar la cuantificación inversa, es decir, la descuantificación, de los bloques de coeficientes asociados a la TU. La unidad de cuantificación inversa 154 puede usar un valor de QP asociado a la CU de la TU para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para que la unidad de cuantificación inversa 154 lo aplique. Es decir, la relación de compresión, es decir, la relación del número de bits usados para representar una secuencia original y la secuencia comprimida, se puede controlar ajustando el valor del QP usado al cuantificar coeficientes de transformada. La relación de compresión también puede depender del procedimiento de codificación de entropía empleado.

15 **[0129]** Después de que la unidad de cuantificación inversa 154 haya realizado la cuantificación inversa de un bloque de coeficientes, la unidad de procesamiento de transformada inversa 156 puede aplicar una o más transformadas inversas al bloque de coeficientes a fin de generar un bloque residual asociado a la TU. Por ejemplo, la unidad de procesamiento de transformada inversa 156 puede aplicar una DCT inversa, una transformada entera inversa, una transformada inversa de Karhunen-Loeve (KLT), una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes.

20 **[0130]** Si se codifica una PU usando intrapredicción, la unidad de procesamiento de intrapredicción 166 puede realizar una intrapredicción para generar bloques predictivos para la PU. La unidad de procesamiento de intrapredicción 166 puede usar un modo de intrapredicción para generar los bloques predictivos de luma, Cb y Cr para la PU en base a los bloques de predicción de unas PU vecinas en el espacio. La unidad de procesamiento de intrapredicción 166 puede determinar el modo de intrapredicción para la PU en base a uno o más elementos sintácticos descodificados de la corriente de bits.

30 **[0131]** La unidad de procesamiento de predicción 152 puede construir una primera lista de imágenes de referencia (RefPicList0) y una segunda lista de imágenes de referencia (RefPicList1) en base unos elementos sintácticos extraídos de la corriente de bits. Además, si se codifica una PU usando interpredicción, la unidad de descodificación de entropía 150 puede extraer información de movimiento para la PU. La unidad de compensación de movimiento 164 puede determinar, en base a 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, en base a unos bloques de muestras del uno o más bloques de referencia para la PU, bloques predictivos de luma, Cb y Cr para la PU.

35 **[0132]** La unidad de reconstrucción 158 puede usar los bloques de transformada de luma, Cb y Cr, asociados a las TU de una CU y los bloques predictivos de luma, Cb y Cr de las PU de la CU, es decir, datos de intrapredicción o datos de interpredicción, según corresponda, para reconstruir los bloques de codificación de luma, Cb y Cr de la CU. Por ejemplo, la unidad de reconstrucción 158 puede añadir muestras de los bloques de transformada de luma, Cb y Cr a unas muestras correspondientes de los bloques predictivos de luma, Cb y Cr para reconstruir los bloques de codificación de luma, Cb y Cr de la CU.

45 **[0133]** La unidad de filtro 160 puede realizar una operación antibloque para reducir los artefactos de bloque asociados a los bloques de codificación de luma, Cb y Cr de la CU. El descodificador de vídeo 30 puede almacenar los bloques de codificación de luma, Cb y Cr de la CU en la memoria intermedia de imágenes descodificadas 162. La memoria intermedia de imágenes descodificadas 162 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intrapredicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, en base a los bloques de luma, Cb y Cr de la memoria intermedia de imágenes descodificadas 162, operaciones de intrapredicción o de interpredicción en las PU de otras CU. De esta manera, el descodificador de vídeo 30 puede extraer, a partir de la corriente de bits, unos niveles de coeficientes de transformada del bloque de coeficientes de luma significativos, realizar la cuantificación inversa de los niveles de coeficientes de transformada, aplicar una transformada a los niveles de coeficientes de transformada para generar un bloque de transformada, generar, en base, al menos en parte, al bloque de transformada, un bloque de codificación, y facilitar el bloque de codificación para su visualización.

55 **[0134]** De acuerdo con diversos ejemplos de esta 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 basada en paleta, por ejemplo, para una CU o PU. Por ejemplo, la unidad de descodificación basada en paleta 165 puede estar configurada para generar una paleta que tiene entradas que indican valores de píxel, recibir información que asocia al menos algunas posiciones de un bloque de datos de vídeo a entradas de la paleta, seleccionar valores de píxel de la paleta en base a la información y reconstruir valores de píxel del bloque en base a los valores de píxel seleccionados. Aunque se describe que la unidad de descodificación basada en paleta 165 realiza diversas funciones, otras unidades de procesamiento o una combinación de diferentes unidades de procesamiento pueden realizar algunas o la totalidad de dichas funciones.

65

**[0135]** La unidad de descodificación basada en paleta 165 puede recibir información de modo de codificación de paleta, y realizar las operaciones anteriores cuando la información de 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 de modo indica el uso de un modo diferente, la unidad de procesamiento de predicción 152 descodifica el bloque de datos de vídeo usando un modo de codificación no basada en paleta, por ejemplo, tal como un modo HEVC interpredictivo usando la unidad de compensación de movimiento 164 o un modo de codificación intrapredictivo usando la unidad de procesamiento de intrapredicción 166, cuando la información de 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 codificación de predicción temporal interpredictiva o espacial intrapredictiva y descodificar otros bloques con el modo de codificación basada en paleta. El modo de codificación basada en paleta puede comprender uno de una pluralidad de diferentes modos de codificación basadas en paleta, o puede haber un único modo de codificación basada en paleta.

**[0136]** La información de modo de codificación de paleta, recibida por la unidad de descodificación basada en paleta 165, puede comprender un elemento sintáctico de modo de paleta, tal como una marca. Un primer valor del elemento sintáctico de modo de paleta indica que el modo de codificación de paleta se aplica al bloque y un segundo valor del elemento sintáctico de modo de paleta indica que el modo de codificación de paleta no se aplica al bloque de datos de vídeo. La unidad de descodificación basada en paleta 165 puede recibir la información de modo de codificación de paleta en uno o más de un nivel de unidad predictiva, un nivel de unidad de codificación, un nivel de segmento o un nivel de imagen, o puede recibir la información de modo de codificación de paleta en al menos uno de un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de secuencia (SPS) o un conjunto de parámetros de vídeo (VPS).

**[0137]** En algunos ejemplos, la unidad de descodificación basada en paleta 165 puede deducir la información de modo de codificación de paleta en base a uno o más de un tamaño del bloque de codificación, un tipo de trama, un espacio de color, un componente de color, un tamaño de trama, una velocidad de trama, un ID de capa en codificación de vídeo escalable o un ID de vista en codificación multivista asociado al bloque de datos de vídeo.

**[0138]** La unidad de descodificación basada en paleta 165 también puede estar configurada para recibir información que define al menos algunas de las entradas de la paleta con datos de vídeo, y generar la paleta en base, al menos en parte, a la información recibida. El tamaño de la paleta puede ser fijo o variable. En algunos casos, el tamaño de la paleta es variable y es ajustable en base a la información señalada con los datos de vídeo. La información señalada puede especificar si una entrada de la paleta es una última entrada de la paleta. Además, en algunos casos, la paleta puede tener un tamaño máximo.

**[0139]** La paleta puede ser una paleta única que incluye entradas que indican valores de píxel para un componente de luma y unos componentes de croma del bloque. En este caso, cada entrada de la paleta es una entrada triple que indica valores de píxel para el componente de luma y dos componentes de croma. De forma alternativa, la paleta comprende una paleta de luma que incluye entradas que indican valores de píxel de un componente de luma del bloque, y unas paletas de croma que incluyen entradas que indican valores de píxel para unos respectivos componentes de croma del bloque.

**[0140]** En algunos ejemplos, la unidad de descodificación basada en paleta 165 puede generar la paleta prediciendo las entradas de la paleta en base a unos datos procesados previamente. Los datos procesados previamente pueden incluir paletas, o información de paletas, para bloques vecinos descodificados previamente. La unidad de descodificación basada en paleta 165 puede recibir un elemento sintáctico de predicción que indica si las entradas de la paleta se deben predecir. El elemento sintáctico de predicción puede incluir una pluralidad de elementos sintácticos de predicción que indican, respectivamente, si unas entradas de las paletas para componentes de luma y croma se deben predecir.

**[0141]** La unidad de descodificación basada en paleta 165 puede, en algunos ejemplos, predecir al menos algunas de las entradas de la paleta en base a unas entradas de una paleta para un bloque vecino izquierdo o un bloque vecino superior en un segmento o imagen. En este caso, las entradas de la paleta que se predicen en base a unas entradas de una paleta para el bloque vecino izquierdo o bien el bloque vecino superior se pueden predecir mediante la unidad de descodificación basada en paleta 165, en base a un elemento sintáctico que indica una selección del bloque vecino izquierdo o el bloque vecino superior para su predicción. El elemento sintáctico puede ser una marca que tiene un valor que indica una selección del bloque vecino izquierdo o el bloque vecino superior para su predicción.

**[0142]** En algunos ejemplos, la unidad de descodificación basada en paleta 165 puede recibir uno o más elementos sintácticos de predicción que indican si al menos algunas entradas seleccionadas de la paleta, de entrada en entrada, se deben predecir, y generar las entradas como corresponda. La unidad de descodificación basada en paleta 165 puede predecir algunas de las entradas y recibir información que especifica directamente otras entradas de la paleta.

- 5 **[0143]** La información, recibida por la unidad de descodificación basada en paleta 165, que asocia al menos algunas posiciones de un bloque de datos de vídeo a entradas de la paleta, puede comprender información de mapa que incluye valores de índice de paleta para al menos algunas de las posiciones en el bloque, en el que cada uno de los valores de índices de paleta corresponde a una de las entradas de la paleta. La información de mapa puede incluir uno o más elementos sintácticos en sucesión, cada uno de los cuales indica un número de posiciones consecutivas en el bloque que tienen el mismo valor de índice de paleta.
- 10 **[0144]** En algunos ejemplos, la unidad de descodificación basada en paleta 165 puede recibir información que indica una copia de líneas, con la que unas entradas de paleta para una línea de posiciones en el bloque se copian de unas entradas de paleta para otra línea de posiciones en el bloque. La unidad de descodificación basada en paleta 165 puede usar esta información para realizar una copia de líneas para determinar unas entradas de la paleta para diversas posiciones de un bloque. La línea de posiciones puede comprender una fila, una parte de una fila, una columna o una parte de una columna de posiciones del bloque.
- 15 **[0145]** La unidad de descodificación basada en paleta 165 puede generar la paleta en parte recibiendo valores de píxel para una o más posiciones del bloque, y añadiendo los valores de píxel a unas entradas de la paleta para generar dinámicamente al menos una parte de la paleta al instante. Añadir los valores de píxel puede comprender añadir los valores de píxel a una paleta inicial que comprende un conjunto inicial de entradas, o a una paleta vacía que no incluye un conjunto inicial de entradas. En algunos ejemplos, añadir comprende añadir los valores de píxel para añadir nuevas entradas a una paleta inicial que comprende un conjunto inicial de entradas o rellenar entradas existentes en la paleta inicial, o reemplazar o cambiar valores de píxel de unas entradas en la paleta inicial.
- 20 **[0146]** En algunos ejemplos, la paleta puede ser una paleta cuantificada en la que un valor de píxel seleccionado de la paleta para una de las posiciones en el bloque es diferente a un valor de píxel real de la posición en el bloque, de modo que el proceso de descodificación tiene pérdidas. Por ejemplo, se puede seleccionar el mismo valor de píxel de la paleta para dos posiciones diferentes que tienen diferentes valores de píxel reales.
- 25 **[0147]** Como se analiza anteriormente, la unidad de descodificación basada en paleta 165 puede recibir información que define una paleta para un bloque actual de datos de vídeo. Por ejemplo, la unidad de descodificación basada en paleta 165 puede recibir una pluralidad de elementos sintácticos desde la unidad de descodificación de entropía 150. En algunos ejemplos, la unidad de descodificación de entropía 150 puede descodificar la pluralidad de elementos sintácticos a partir de una corriente de bits de vídeo codificada de acuerdo con una tabla de sintaxis. Como ejemplo, la unidad de descodificación de entropía 150 puede descodificar la pluralidad de elementos sintácticos a partir de una corriente de bits de vídeo codificada de acuerdo con la tabla de sintaxis de paleta del borrador 3 de SCC HEVC, que se reproduce anteriormente en la Tabla 1. Sin embargo, como se analiza anteriormente, la disposición de elementos sintácticos del borrador 3 de SCC HEVC puede no ser óptima. En particular, la disposición de elementos sintácticos en el borrador 3 de SCC HEVC no aumenta al máximo el número de elementos sintácticos codificados en modo de derivación que se agrupan, lo que puede disminuir el rendimiento CABAC.
- 30 **[0148]** De acuerdo con una o más técnicas de esta divulgación, la unidad de descodificación de entropía 150 puede descodificar los elementos sintácticos usados para definir una paleta actual de modo que unos elementos sintácticos adicionales codificados en modo de derivación se agrupen juntos. Por ejemplo, en lugar de separar los períodos de los elementos sintácticos **palette\_predictor\_run**, **num\_signalled\_palette\_entries**, **palette\_entry** y **palette\_escape\_val\_present\_flag** y los períodos de los elementos sintácticos **num\_palette\_indices\_idc**, y **palette\_index\_idc**, la unidad de descodificación de entropía 150 puede descodificar uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo después de un elemento sintáctico que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de vídeo, de modo que los períodos de los elementos sintácticos **palette\_predictor\_run**, **num\_signalled\_palette\_entries**, **palette\_entry** y **palette\_escape\_val\_present\_flag**, **num\_palette\_indices\_idc** y **palette\_index\_idc** se agrupan juntos. Como ejemplo, la unidad de descodificación de entropía 150 puede descodificar los elementos sintácticos usados para definir la paleta actual en el orden mostrado anteriormente en la Tabla 4. Como otro ejemplo, la unidad de descodificación de entropía 150 puede descodificar los elementos sintácticos usados para definir la paleta actual en el orden mostrado anteriormente en la Tabla 5. De esta manera, se puede incrementar el rendimiento CABAC de la unidad de descodificación de entropía 150.
- 35 **[0149]** La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de determinación de una paleta para codificar datos de vídeo, consecuente con las técnicas de esta divulgación. El ejemplo de la FIG. 4 incluye una imagen 178 que tiene una primera unidad de codificación (CU) 180 que está asociada a unas primeras paletas 184 y una segunda CU 188 que está asociada a unas segundas paletas 192. Como se describe en mayor detalle a continuación y de acuerdo a las técnicas de esta divulgación, las segundas paletas 192 están basadas en las primeras paletas 184. La imagen 178 también incluye el bloque 196 codificado con un modo de codificación de intrapredicción y el bloque 200 que está codificado con un modo de codificación de interpredicción.
- 40 **[0150]** 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 al estándar HEVC con propósitos explicativos. Sin embargo, se debería entender que las técnicas de esta divulgación no están limitadas de esta manera, y que otros

procesadores y/o dispositivos de codificación de vídeo pueden aplicarlas en otros procesos y/o estándares de codificación de vídeo.

**[0151]** En general, una paleta se refiere a un número de valores de píxel que son dominantes y/o representativos para una CU que se está codificando actualmente, tal como la CU 188 en el ejemplo de la FIG. 4. Las primeras paletas 184 y las segundas paletas 192 mostradas incluyen múltiples paletas. En algunos ejemplos, 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, unas entradas de la paleta Y pueden representar valores Y de píxeles de la CU, unas entradas de la paleta U pueden representar valores U de píxeles de la CU y unas entradas de la paleta V pueden representar valores V de píxeles de la CU. En otro ejemplo, el codificador de vídeo 20 puede codificar una paleta para un componente de luma (Y) de una CU, y otra paleta para dos componentes (U, V) de la CU. En este ejemplo, las entradas de la paleta Y pueden representar valores Y de píxeles de la CU, y las entradas de la paleta U-V pueden representar pares de valores U-V de píxeles de la CU.

**[0152]** En otros ejemplos, el codificador de vídeo 20 puede codificar una única paleta para todos los componentes de color de una CU. En este ejemplo, el codificador de vídeo 20 puede codificar una paleta que tiene una *i*-ésima entrada que es un valor triple, que incluye  $Y_i$ ,  $U_i$  y  $V_i$ . En este caso, la paleta incluye valores para cada uno de los componentes de los píxeles. En consecuencia, la representación de las paletas 184 y 192 como un conjunto de paletas que tiene múltiples paletas individuales es simplemente un ejemplo y no pretende ser limitativo.

**[0153]** En el ejemplo de la FIG. 4, la primera paleta 184 incluye tres entradas 202-206 que tienen un valor de índice de entrada 1, un valor de índice de entrada 2 y un valor de índice de entrada 3, respectivamente. Las entradas 202-206 relacionan los valores de índice con valores de píxel, incluyendo un valor de píxel A, un valor de píxel B y un valor de píxel C, respectivamente. Como se describe en el presente documento, en vez de codificar los valores de píxel reales de la primera CU 180, un codificador de vídeo (tal como un codificador de vídeo 20 o un descodificador de vídeo 30) puede usar la codificación basada en paleta para codificar los píxeles del bloque usando los índices 1-3. Es decir, para cada posición de píxel de la primera CU 180, el codificador de vídeo 20 puede codificar un valor de índice para el píxel, donde el valor de índice está asociado a un valor de píxel en una o más de las primeras paletas 184. El descodificador de vídeo 30 puede obtener los valores de índice a partir de una corriente de bits y reconstruir los valores de píxel usando los valores de índice y una o más de las primeras paletas 184. Por tanto, el codificador de vídeo 20 transmite unas primeras paletas 184 en una corriente de bits de datos de vídeo codificada para su uso por el descodificador de vídeo 30 en la descodificación basada en paleta. En general, se pueden transmitir una o más paletas para cada CU o se pueden compartir entre diferentes CU.

**[0154]** El codificador de vídeo 20 y el descodificador de vídeo 30 pueden determinar unas segundas paletas 192 en base a unas primeras paletas 184. Por ejemplo, el codificador de vídeo 20 puede codificar una `pred_palette_flag` para cada CU (incluyendo, como ejemplo, una segunda CU 188) para indicar si la paleta para la CU se predice a partir de una o más paletas asociadas a una o más CU diferentes, tales como unas CU vecinas (en el espacio o en base a un orden de exploración) o las muestras más frecuentes de un vecino causal. Por ejemplo, cuando el valor de dicha marca 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 una corriente de bits que contiene la `pred_palette_flag`. Cuando dicha marca es igual a cero, el descodificador de vídeo 30 puede determinar que la paleta 192 para la segunda CU 188 está incluida en la corriente de bits como una nueva paleta. En algunos ejemplos, `pred_palette_flag` se puede codificar por separado para cada componente de color diferente de una CU (por ejemplo, tres marcas, una para Y, una para U y una para V, para una CU en vídeo YUV). En otros ejemplos, se puede codificar una única `pred_palette_flag` para todos los componentes de color de una CU.

**[0155]** En el ejemplo anterior, se señala la `pred_palette_flag` para cada CU para indicar si se predice alguna de las entradas de la paleta para el bloque actual. En algunos ejemplos, se puede señalar uno o más elementos sintácticos para cada entrada. Es decir, se puede señalar una marca para cada entrada de un predictor de paleta para indicar si esa entrada está presente en la paleta actual. Como se ha indicado anteriormente, si una entrada de paleta no se predice, la entrada de paleta se puede señalar explícitamente.

**[0156]** Cuando se determinan unas segundas paletas 192 en relación con unas primeras paletas 184 (por ejemplo, `pred_palette_flag` 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 cuales se determinan las paletas predictivas, en este ejemplo, las primeras paletas 184. Las paletas predictivas se pueden asociar a una o más CU vecinas de la CU que se está codificando actualmente (por ejemplo, tales como unas CU vecinas (en el espacio o en base a un orden de exploración) o las muestras más frecuentes de un vecino causal), es decir, la segunda CU 188. Las paletas de la una o más CU vecinas pueden estar asociadas a una paleta predictora. En algunos ejemplos, tales como el ejemplo ilustrado en la FIG. 4, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden localizar una CU vecina izquierda, la primera CU 180, al determinar 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 relativas a la segunda CU 188, tal como una CU superior, la CU 196.

**[0157]** El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar una CU para una predicción de paleta en base a una jerarquía. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden identificar inicialmente la CU vecina izquierda, la primera CU 180, para una predicción de paleta. Si la CU vecina izquierda no está disponible para la predicción (por ejemplo, la CU vecina izquierda está codificada con un modo diferente a un modo de codificación basada en paleta, tal como un modo de intrapredicción o un modo de intrapredicción, o está localizada en el borde más a la izquierda de una imagen o segmento), el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden identificar la CU vecina 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 una predicción de paleta. En algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden determinar una paleta predictiva en base a múltiples bloques y/o muestras reconstruidas de un bloque vecino.

**[0158]** Aunque el ejemplo de la FIG. 4 ilustra unas primeras paletas 184 como paletas predictivas de una única CU, la primera CU 180, en otros ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden localizar paletas para predicción a partir de una combinación de CU vecinas. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo pueden aplicar una o más fórmulas, funciones, reglas o similares para generar una paleta en base a unas paletas de una o una combinación de una pluralidad de CU vecinas.

**[0159]** En otros ejemplos más, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden construir una lista de candidatos que incluye un número de candidatos potenciales para predicción de paletas. Se puede aplicar un proceso de poda tanto en el codificador de vídeo 20 como en el descodificador de vídeo 30 para eliminar candidatos duplicados de la lista. En dichos ejemplos, el codificador de vídeo 20 puede codificar un índice a la lista de candidatos, para indicar la CU candidata de la lista de la cual se selecciona la CU actual usada para la predicción de paleta (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.

**[0160]** En un ejemplo con propósitos 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 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 una marca que tiene un valor de cero para indicar que la paleta para la CU actual está copiada de la CU situada a la izquierda de la CU actual. El codificador de vídeo 20 puede codificar la marca que tiene un valor de uno para indicar que la paleta para la CU actual está copiada de la CU situada encima de la CU actual. El descodificador de vídeo 30 descodifica la marca y selecciona la CU adecuada para la predicción de paleta.

**[0161]** En otros ejemplos más, el codificador de vídeo 20 y/o el descodificador de vídeo 30 determinan la paleta para la CU que se está codificando actualmente en base a la frecuencia con la que unos valores de muestra incluidos en una o más paletas diferentes aparecen en una o más CU vecinas. Por ejemplo, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden efectuar un seguimiento de los colores asociados a 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.

**[0162]** En algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden realizar una predicción de paleta en base a cada entrada. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos, tal como una o más marcas, para cada entrada de una paleta predictiva que indica si las entradas de paleta predictiva respectivas se reúsan en la paleta actual (por ejemplo, si la paleta actual reusa valores de píxel de una paleta de otra CU). En este ejemplo, el codificador de vídeo 20 puede codificar una marca que tiene un valor igual a uno para una entrada dada cuando la entrada es un valor predicho a partir de una paleta predictiva (por ejemplo, una entrada correspondiente de una paleta asociada a una CU vecina). El codificador de vídeo 20 puede codificar una marca que tiene un valor igual a cero para una entrada particular, para indicar que la entrada en 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.

**[0163]** En el ejemplo de la FIG. 4, unas segundas paletas 192 incluyen cuatro entradas 208-214 que tienen un valor de índice de entrada 1, un valor de índice de entrada 2, un valor de índice de entrada 3 y un índice de entrada 4, respectivamente. Las entradas 208-214 relacionan los valores de índice con unos valores de píxel que incluyen el valor de píxel A, el valor de píxel B, el valor de píxel C y el valor de píxel D, respectivamente. 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 propósitos de predicción de paleta y copiar las entradas 1-3 de unas primeras paletas 184 en las entradas 1-3 de unas segundas paletas 192, para codificar una segunda CU 188. De esta manera, el codificador de

vídeo 20 y/o el descodificador de vídeo 30 pueden determinar unas segundas paletas 192 en base a unas primeras paletas 184. Además, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar datos para que la entrada 4 se incluya con las segundas paletas 192. Dicha información puede incluir el número de entradas de paleta no predichas a partir de una paleta predictora y los valores de píxel correspondientes a esas entradas de paleta.

**[0164]** En algunos ejemplos, de acuerdo con unos aspectos de esta divulgación, uno o más elementos sintácticos pueden indicar si unas paletas, tales como unas segundas paletas 192, se predicen por completo a partir de una paleta predictiva (mostrada en la FIG. 4 como unas primeras paletas 184, pero que puede estar compuesta de entradas de uno o más bloques) o si unas entradas particulares de unas segundas paletas 192 se predicen. 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, una marca que tiene un valor de 0), uno o más elementos sintácticos adicionales pueden indicar qué entradas de unas segundas paletas 192 se predicen a partir de la paleta predictiva.

**[0165]** De acuerdo con algunos aspectos de esta divulgación, determinada información asociada a la predicción de paletas se puede deducir a partir de una o más características de los datos que se codifican. Es decir, en lugar de que el codificador de vídeo 20 codifique elementos sintácticos (y que el descodificador de vídeo 30 descodifique dichos elementos sintácticos), el codificador de vídeo 20 y el descodificador de vídeo 30 pueden realizar una predicción de paleta en base a una o más características de los datos que se codifican.

**[0166]** La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de determinación de índices a una paleta para un bloque de píxeles, consecuente con las técnicas de esta divulgación. Por ejemplo, la FIG. 5 incluye un mapa 240 de valores de índice (valores 1, 2 y 3) que relacionan unas posiciones respectivas de píxeles asociados a los valores de índices con una entrada de unas paletas 244. Las paletas 244 se pueden determinar de manera similar a las primeras paletas 184 y las segundas paletas 192 descritas anteriormente con respecto a la FIG. 4.

**[0167]** De nuevo, 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 al estándar de codificación de vídeo HEVC, con propósitos explicativos. Sin embargo, se debería entender que las técnicas de esta divulgación no están limitadas de esta manera, y que otros procesadores y/o dispositivos de codificación de vídeo pueden aplicarlas en otros procesos y/o estándares de codificación de vídeo.

**[0168]** Aunque el mapa 240 que se ilustra en el ejemplo de la FIG. 5 incluye un valor de índice para cada posición de píxel, se debería entender que, en otros ejemplos, no todas las posiciones de píxel pueden estar asociadas a un valor de índice que relaciona el valor de píxel con una entrada de las paletas 244. Es decir, como se indica anteriormente, en algunos ejemplos, el codificador de vídeo 20 puede codificar (y el descodificador de vídeo 30 puede obtener, a partir de una corriente de bits codificada) una indicación de un valor de píxel real (o su versión cuantificada) para una posición en el mapa 240 si el valor de píxel no está incluido en las paletas 244.

**[0169]** En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar configurados para codificar un mapa adicional que indica qué posiciones de píxel están asociadas a valores de índice. Por ejemplo, se va a suponer que la entrada (i, j) del 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) que indican si la entrada tiene un valor de índice asociado. Por ejemplo, el codificador de vídeo 20 puede codificar una marca 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 de 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 de la paleta y permitir que el descodificador de vídeo reconstruya el valor de píxel. En casos en los que las paletas 244 incluyen una sola entrada y un valor de píxel asociado, el codificador de vídeo 20 puede omitir la señalización del valor de índice. El codificador de vídeo 20 puede codificar la marca para que tenga un valor de cero para indicar que el valor de píxel en la localización (i, j) de la CU no es uno de los valores de las paletas 244. En este ejemplo, el codificador de vídeo 20 también puede codificar una indicación del valor de píxel para su uso por el descodificador de vídeo 30 al reconstruir el valor de píxel. En algunos casos, el valor de píxel se puede codificar de una manera que conlleva pérdidas.

**[0170]** El valor de un píxel en una posición de una CU puede proporcionar una indicación de valores de uno o más píxeles diferentes en otras posiciones de la CU. Por ejemplo, puede haber una probabilidad relativamente alta de que unas posiciones de píxeles vecinos de una CU tengan el mismo valor de píxel o puedan estar mapeadas al mismo valor de índice (en el caso de codificación con pérdidas, en la que más de un valor de píxel puede estar mapeado a un único valor de índice).

**[0171]** En consecuencia, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos que indican un número de píxeles o valores de índice consecutivos en un orden de exploración dado que tienen el mismo valor de píxel o valor de índice. Como se indica anteriormente, la cadena de valores de píxel o índice de valor similar se puede denominar recorrido en el presente documento. En un ejemplo con propósitos ilustrativos, si dos píxeles o índices consecutivos en un orden de exploración dado tienen valores diferentes, el recorrido 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, el recorrido es igual a uno. Para tres índices o píxeles consecutivos con el mismo valor, el recorrido es dos, etc. El descodificador de vídeo 30 puede obtener los elementos sintácticos que indican un recorrido a partir de una corriente de bits codificada, y usar los datos para determinar el número de localizaciones consecutivas que tienen el mismo valor de píxel o índice.

5 **[0172]** El orden de exploración puede afectar al número de índices que pueden estar incluidos en un recorrido. Por ejemplo, se va a considerar una exploración de barrido de las líneas 266, 268 y 270 del mapa 240. Si se supone una dirección de exploración horizontal de izquierda a derecha (tal como un orden de exploración de barrido), la fila 266 incluye tres valores de índice de "1", dos valores de índice de "2" y tres valores de índice de "3". La fila 268 incluye cinco valores de índice de "1" y tres valores de índice de "3". En este ejemplo, para la fila 266, el codificador de vídeo 20 puede codificar elementos sintácticos que indican que el primer valor de la fila 266 (el valor más a la izquierda de la fila) es 1 con un recorrido de 2, seguido de un valor de índice de 2 con un recorrido de 1, seguido de un valor de índice de 3 con un recorrido de 2. Después de la exploración de barrido, el codificador de vídeo 20 puede comenzar a codificar la fila 268 con el valor más a la izquierda. Por ejemplo, el codificador de vídeo 20 puede codificar elementos sintácticos que indican que el primer valor de la fila 268 es 1 con un recorrido de 4, seguido de un valor de índice de 3 con un recorrido de 2. El codificador de vídeo 20 puede proceder de la misma manera con la línea 270.

20 **[0173]** Por consiguiente, en el orden de exploración de barrido, el primer índice de una línea actual se puede explorar directamente después del último índice de una línea previa. Sin embargo, en algunos ejemplos, puede no ser deseable explorar los índices en un orden de exploración de barrido. Por ejemplo, puede no ser deseable explorar los índices en un orden de exploración de barrido donde una primera línea de un bloque de datos de vídeo (por ejemplo, la fila 266) incluye un primer píxel adyacente a un primer borde del bloque de datos de vídeo (por ejemplo, el píxel más a la izquierda de la fila 266, que tiene un valor de índice de 1) y un último píxel adyacente a un segundo borde del bloque de datos de vídeo (por ejemplo, el píxel más a la derecha de la fila 266, que tiene un valor de índice de 3), una segunda línea del bloque de datos de vídeo (por ejemplo, la fila 268) incluye un primer píxel adyacente al primer borde del bloque de datos de vídeo (por ejemplo, el píxel más a la izquierda de la fila 268, que tiene un valor de índice de 1) y un último píxel adyacente al segundo borde del bloque de datos de vídeo (por ejemplo, el píxel más a la derecha de la fila 268, que tiene un valor de índice de 3), el último píxel de la primera línea es adyacente al último píxel de la segunda línea, y el primer borde y el segundo borde son paralelos, y el último píxel de la primera línea tiene el mismo valor de índice que el último píxel de la segunda línea, pero tiene un valor de índice diferente del primer píxel de la segunda línea. Esta situación (es decir, cuando el valor de índice del último píxel de la primera línea es igual al último píxel de la segunda línea, pero diferente del primer píxel de la segunda línea) se puede producir con más frecuencia en el contenido de pantalla generado por ordenador que en otros tipos de contenido de vídeo.

35 **[0174]** En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración serpenteante cuando codifica los índices del mapa. Por ejemplo, el codificador de vídeo 20 puede explorar el último píxel de la segunda línea directamente después del último píxel de la primera línea. De esta manera, el codificador de vídeo 20 puede mejorar la eficacia de la codificación de longitud de recorrido.

40 **[0175]** Por ejemplo, en lugar de usar un orden de exploración de barrido, el codificador de vídeo 20 puede usar un orden de exploración serpenteante para codificar los valores del mapa 240. En un ejemplo con propósitos ilustrativos, se van a considerar las filas 266, 268 y 270 del mapa 240. Usando un orden de exploración serpenteante (tal como un orden de exploración en zigzag), el codificador de vídeo 20 puede codificar los valores del mapa 240 comenzando por la posición izquierda de la fila 266, continuando hasta la posición más a la derecha de la fila 266, descendiendo hasta la posición más a la izquierda de la fila 268, continuando hasta la posición más a la izquierda de la fila 268, y descendiendo hasta la posición más a la izquierda de la fila 270. Por ejemplo, el codificador de vídeo 20 puede codificar uno o más elementos sintácticos que indican que la primera posición de la fila 266 es uno y que el siguiente recorrido de dos entradas consecutivas en el orden de exploración es el mismo que la primera posición de la fila 266.

50 **[0176]** El codificador de vídeo 20 puede codificar uno o más elementos sintácticos que indican que la siguiente posición de la fila 266 (es decir, la cuarta posición, de izquierda a derecha) es dos y que la siguiente entrada consecutiva en la dirección de exploración es la misma que la cuarta posición de la fila 266. El codificador de vídeo 20 puede codificar uno o más elementos sintácticos que indican que la siguiente posición de la fila 266 (es decir, la sexta posición) es tres y que el próximo recorrido de cinco entradas consecutivas en la dirección de exploración es el mismo que la sexta posición de la fila 266. El codificador de vídeo 20 puede codificar uno o más elementos sintácticos que indican que la siguiente posición en la dirección de exploración (es decir, la cuarta posición de la fila 268, de derecha a izquierda) de la fila 268 es uno y que el próximo recorrido de nueve entradas consecutivas en la dirección de exploración es el mismo que la cuarta posición de la fila 268.

60 **[0177]** De esta manera, usando un orden de exploración serpenteante, el codificador de vídeo 20 puede codificar recorridos de mayor longitud, lo que puede mejorar la eficacia de codificación. Por ejemplo, usando la exploración de barrido, el recorrido final de la fila 266 (para el valor de índice 3) es igual a 2. Sin embargo, usando la exploración serpenteante, el recorrido final de la fila 266 se extiende a la fila 268 y es igual a 5.

65 **[0178]** El descodificador de vídeo 30 puede recibir los elementos sintácticos descritos anteriormente y puede reconstruir las filas 266, 268 y 270. Por ejemplo, el descodificador de vídeo 30 puede obtener, a partir de una corriente

de bits codificada, datos que indican un valor de índice para una posición de mapa 240 que se está 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.

5 **[0179]** La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de proceso para descodificar un bloque de datos de vídeo usando un modo de paleta, de acuerdo con una o más técnicas de esta divulgación. Un descodificador de vídeo, tal como el descodificador de vídeo 30 ilustrado en la FIG. 1 y la FIG. 3, puede realizar las técnicas de la FIG. 6. Para propósitos ilustrativos, las técnicas de la FIG. 6 se describen dentro del contexto del descodificador de vídeo 30 de la FIG. 1 y la FIG. 3, aunque los descodificadores de vídeo que tienen configuraciones diferentes a la del descodificador de vídeo 30 pueden realizar las técnicas de la FIG. 6.

15 **[0180]** Como se analiza anteriormente, puede ser deseable aumentar al máximo el número de períodos codificados en modo de derivación de elementos sintácticos que se agrupan. De acuerdo con una o más técnicas de esta divulgación, el descodificador de vídeo 30 puede descodificar, a partir de una corriente de bits de vídeo codificada y usando el modo de derivación, un grupo de elementos sintácticos para una paleta para un bloque actual de datos de vídeo (602). Por ejemplo, la unidad de descodificación de entropía 150 del descodificador de vídeo 30 puede descodificar, usando un modo de derivación, períodos de uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si las entradas de una paleta predictora se reúsan en la paleta actual (por ejemplo, uno o más elementos sintácticos **palette\_predictor\_run**), un elemento sintáctico que indica un número de entradas de la paleta actual que están señaladas explícitamente (por ejemplo, un elemento sintáctico **num\_signalled\_palette\_entries**), uno o más elementos sintácticos que indican cada uno un valor de un componente de una entrada de la paleta actual (por ejemplo, uno o más elementos sintácticos **palette\_entry**), un elemento sintáctico que indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape (por ejemplo, un elemento sintáctico **palette\_escape\_val\_present\_flag**), un elemento sintáctico que indica un número de entradas de la paleta actual que se señalan o deducen explícitamente (por ejemplo, un elemento sintáctico **num\_palette\_indices\_idc**), y uno o más elementos sintácticos que indican índices de una matriz de entradas de paleta actual (por ejemplo, uno o más elementos sintácticos **palette\_index\_idc**). En algunos ejemplos, para descodificar un grupo de elementos sintácticos codificados en derivación, el descodificador de vídeo 30 puede descodificar en secuencia elementos sintácticos incluidos en el grupo de elementos sintácticos sin descodificar ningún período no codificado en derivación. Como se analiza anteriormente, agrupar un gran número de períodos/elementos sintácticos/codificados en derivación puede mejorar el rendimiento CABAC del descodificador de vídeo 30. En particular, la agrupación de elementos sintácticos codificados en derivación puede permitir que el descodificador de vídeo 30 evite arrancar/detener/volver a arrancar el motor CABAC. Por el contrario, cuando los elementos sintácticos codificados en derivación no están agrupados, el descodificador de vídeo 30 puede tener que arrancar continuamente el motor CABAC para descodificar un período no codificado en derivación con un primer contexto, detener el motor CABAC para descodificar un período codificado en derivación, arrancar el motor CABAC para descodificar otro período no codificado en derivación con el primer contexto, etc. Como se analiza anteriormente, la alternancia repetida del motor CABAC puede disminuir el rendimiento del motor CABAC.

40 **[0181]** El descodificador de vídeo 30 puede descodificar, usando CABAC con un contexto y en una posición en la corriente de bits de vídeo codificada que está después del grupo de elementos sintácticos, un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo (604). Por ejemplo, la unidad de descodificación de entropía 150 del descodificador de vídeo 30 puede descodificar, usando CABAC con un contexto, el período de un elemento sintáctico **palette\_transpose\_flag**.

45 **[0182]** El descodificador de vídeo 30 puede descodificar, usando CABAC con un contexto y en una posición en la corriente de bits de vídeo codificada que está después del elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo, uno o más elementos sintácticos relacionados con el parámetro de cuantificación (QP) delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo (606). Por ejemplo, la unidad de descodificación de entropía 150 del descodificador de vídeo 30 puede descodificar, usando CABAC con uno o más contextos, unos períodos de elementos sintácticos que especifican el valor absoluto de una diferencia entre un QP (por ejemplo, un QP de luma) para el bloque actual de datos de vídeo y un predictor del QP para el bloque actual (por ejemplo, **cu\_qp\_delta\_abs**), un elemento sintáctico que especifica un signo de la diferencia entre el QP para el bloque actual de datos de vídeo y el predictor del QP para el bloque actual (por ejemplo, **cu\_qp\_delta\_sign\_flag**), un elemento sintáctico que indica si las entradas de una o más listas de desplazamiento se añaden a un QP de luma para el bloque actual para determinar unos QP de croma para el bloque actual (por ejemplo, **cu\_chroma\_qp\_offset\_flag**) y un elemento sintáctico que especifica un índice de una entrada en cada una de la una o más listas de desplazamiento que se añaden al QP de luma para el bloque actual para determinar los QP de croma para el bloque actual (por ejemplo, **cu\_chroma\_qp\_offset\_idx**).

60 **[0183]** En algunos ejemplos, el descodificador de vídeo 30 puede descodificar el uno o más elementos sintácticos relacionados con el QP delta y/o desplazamientos de QP de croma para el bloque actual de datos de vídeo en base a un valor de un elemento sintáctico del grupo de elementos sintácticos descodificados usando el modo de derivación. Como ejemplo, el descodificador de vídeo 30 puede descodificar el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo donde el elemento

5 sintáctico del grupo de elementos sintácticos indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape indica que el bloque actual de datos de vídeo sí incluye al menos una muestra de escape. Como otro ejemplo, el descodificador de vídeo 30 puede no descodificar el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo donde el elemento sintáctico del grupo de elementos sintácticos que indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape indica que el bloque actual de datos de vídeo no incluye al menos una muestra de escape.

10 [0184] El descodificador de vídeo 30 puede generar la paleta para el bloque actual de datos de vídeo en base al grupo de elementos sintácticos y el elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo (608) y descodifica el bloque actual de datos de vídeo en base a la paleta generada y el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo (610). Por ejemplo, la unidad de descodificación basada en paleta 165 puede generar la paleta que tiene entradas que indican valores de píxel, recibir información que asocia al menos algunas posiciones del bloque actual de datos de vídeo a unas entradas de la paleta, seleccionar valores de píxel en la paleta en base a la información y reconstruir valores de píxel del bloque en base a los valores de píxel seleccionados.

20 [0185] La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de proceso para codificar un bloque de datos de vídeo usando un modo de paleta, de acuerdo con una o más técnicas de esta divulgación. Un codificador de vídeo, tal como el codificador de vídeo 20 ilustrado en la FIG. 1 y la FIG. 2, puede realizar las técnicas de la FIG. 7. Para propósitos ilustrativos, las técnicas de la FIG. 7 se describen dentro del contexto del codificador de vídeo 20 de la FIG. 1 y la FIG. 2, aunque los codificadores de vídeo que tienen configuraciones diferentes a la del codificador de vídeo 20 pueden realizar las técnicas de la FIG. 7.

25 [0186] Como se analiza anteriormente, puede ser deseable aumentar al máximo el número de períodos codificados en modo de derivación de elementos sintácticos que se agrupan. De acuerdo con una o más técnicas de esta divulgación, el codificador de vídeo 20 puede codificar, en una corriente de bits de vídeo codificada y usando el modo de derivación, un grupo de elementos sintácticos para una paleta para un bloque actual de datos de vídeo (702). Por ejemplo, la unidad de codificación de entropía 118 del codificador de vídeo 20 puede codificar, usando un modo de derivación, unos períodos de uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si las entradas de una paleta predictora se reúsan en la paleta actual (por ejemplo, uno o más elementos sintácticos **palette\_predictor\_run**), un elemento sintáctico que indica un número de entradas de la paleta actual que se señalan explícitamente (por ejemplo, un elemento sintáctico **num\_signalled\_palette\_entries**), uno o más elementos sintácticos que indican cada uno un valor de un componente de una entrada de la paleta actual (por ejemplo, uno o más elementos sintácticos **palette\_entry**), un elemento sintáctico que indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape (por ejemplo, un elemento sintáctico **palette\_escape\_val\_present\_flag**), un elemento sintáctico que indica un número de entradas de la paleta actual que se señalan o deducen explícitamente (por ejemplo, un elemento sintáctico **num\_palette\_indices\_idc** o **num\_palette\_indices\_minus1**), y uno o más elementos sintácticos que indican índices de una matriz de entradas de paleta actual (por ejemplo, uno o más elementos sintácticos **palette\_index\_idc**).

45 [0187] El codificador de vídeo 20 puede codificar, usando CABAC con un contexto y en una posición en la corriente de bits de vídeo codificada que está después del grupo de elementos sintácticos, un elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo (704). Por ejemplo, la unidad de codificación de entropía 118 del codificador de vídeo 20 puede codificar, usando CABAC con un contexto, el período de un elemento sintáctico **palette\_transpose\_flag**.

50 [0188] El codificador de vídeo 20 puede codificar, usando CABAC con un contexto y en una posición en la corriente de bits de vídeo codificada que está después del elemento sintáctico que indica si un proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo, uno o más elementos sintácticos relacionados con un parámetro de cuantificación (QP) delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo (706). Por ejemplo, la unidad de codificación de entropía 118 del codificador de vídeo 20 puede codificar, usando CABAC con uno o más contextos, unos períodos de unos elementos sintácticos que especifican el valor absoluto de una diferencia entre un QP de luma para el bloque actual de datos de vídeo y un predictor del QP de luma para el bloque actual (por ejemplo, **cu\_qp\_delta\_abs**), un elemento sintáctico que especifica un signo de la diferencia entre el QP de luma para el bloque actual de datos de vídeo y el predictor del QP de luma para el bloque actual (por ejemplo, **cu\_qp\_delta\_sign\_flag**), un elemento sintáctico que indica si las entradas de una o más listas de desplazamiento se añaden al QP de luma para el bloque actual para determinar unos QP de croma para el bloque actual (por ejemplo, **cu\_chroma\_qp\_offset\_flag**), y un elemento sintáctico que especifica un índice de una entrada de cada una de la una o más listas de desplazamiento que se añaden al QP de luma para el bloque actual para determinar unos QP de croma para el bloque actual (por ejemplo, **cu\_chroma\_qp\_offset\_idx**).

65 [0189] En algunos ejemplos, el codificador de vídeo 20 puede codificar el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo en base a un valor de un elemento sintáctico del grupo de elementos sintácticos codificados usando el modo de

derivación. Como ejemplo, el codificador de vídeo 20 puede codificar el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo donde el elemento sintáctico del grupo de elementos sintácticos que indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape indica que el bloque actual de datos de vídeo sí incluye al menos una muestra de escape. Como otro ejemplo, el codificador de vídeo 20 puede no codificar el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo donde el elemento sintáctico del grupo de elementos sintácticos que indica si el bloque actual de los datos de vídeo incluye al menos una muestra codificada de escape indica que el bloque actual de datos de vídeo no incluye al menos una muestra de escape.

**[0190]** Se debe reconocer que, dependiendo del ejemplo, determinados actos o acontecimientos de cualquiera de las técnicas descritas en el presente documento se pueden realizar en una secuencia distinta, se pueden añadir, combinar o excluir por completo (por ejemplo, no todos los actos o acontecimientos descritos son necesarios para la puesta en práctica de las técnicas). Por otro lado, en determinados ejemplos, los actos o acontecimientos pueden tener lugar simultáneamente, por ejemplo, a través de procesamiento de múltiples subprocesos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente. Además, aunque con propósitos de claridad se describe que un único módulo o unidad realiza determinados aspectos de la presente divulgación, se debería entender que una combinación de unidades o módulos asociados a un codificador de vídeo pueden realizar las técnicas de la presente divulgación.

**[0191]** Determinados aspectos de esta divulgación se han descrito con respecto al estándar de HEVC en desarrollo, con propósitos ilustrativos. Sin embargo, las técnicas descritas en esta divulgación pueden ser útiles para otros procesos de codificación de vídeo, que incluyen otros procesos de codificación de vídeo, estándar o patentados, aún no desarrollados.

**[0192]** El codificador de vídeo 20 (FIGS. 1 y 2) y/o el descodificador de vídeo 30 (FIGS. 1 y 3), ambos de los cuales se pueden denominar en general codificadores de vídeo, pueden realizar las técnicas descritas anteriormente. Del mismo modo, la codificación de vídeo se puede referir a una codificación de vídeo o una descodificación de vídeo, según corresponda.

**[0193]** Aunque se han descrito anteriormente combinaciones particulares de diversos aspectos de las técnicas, estas combinaciones se proporcionan meramente para ilustrar ejemplos de las técnicas descritas en esta divulgación. En consecuencia, las técnicas de esta divulgación no se deberían limitar a estos ejemplos de combinaciones, sino que pueden abarcar cualquier combinación concebible de los diversos aspectos de las técnicas descritas en esta divulgación.

**[0194]** En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, un programa informático, firmware o cualquier combinación de los mismos. Si se implementan en un programa informático, las funciones se pueden almacenar en o transmitirse por un medio legible por ordenador como una o más instrucciones o códigos y ejecutar mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que son no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

**[0195]** A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria *flash* o cualquier otro medio que se pueda usar para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debería entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se dirigen a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, de los cuales el disco flexible normalmente reproduce datos magnéticamente, mientras que los demás discos reproducen datos ópticamente con láseres. Las combinaciones de los anteriores se deberían incluir también dentro del alcance de los medios legibles por ordenador.

5 **[0196]** 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 lógicas programables *in situ* (FPGA) u otros circuitos lógicos integrados o discretos equivalentes pueden ejecutar las instrucciones. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar en módulos de hardware y/o programa informático dedicados configurados para codificar y descodificar, o incorporados en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

15 **[0197]** Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores, como se ha descrito anteriormente, conjuntamente con un programa informático y/o firmware adecuados.

20 **[0198]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

**REIVINDICACIONES**

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

5 descodificar, a partir de una corriente de bits de vídeo codificada y usando codificación aritmética binaria adaptativa a contexto (CABAC) con un contexto, un elemento sintáctico, `palette_transpose_flag`, que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para un bloque actual de datos de vídeo;

10 descodificar, a partir de la corriente de bits de vídeo codificada, usando CABAC con un contexto y en una posición en la corriente de bits de vídeo codificada que está directamente después de la `palette_transpose_flag`, uno o más elementos sintácticos relacionados con un parámetro de cuantificación (QP) delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo a fin de mejorar el rendimiento CABAC;

15 descodificar, a partir de la corriente de bits de vídeo codificada, un grupo de elementos sintácticos consecutivos usando el modo de derivación, en el que el grupo comprende:

20 uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si unas entradas de una paleta predictora se reúsan en la paleta actual;

un elemento sintáctico que indica un número de entradas de la paleta actual que se señalan explícitamente;

25 uno o más elementos sintácticos que indican cada uno un valor de un componente de una entrada de la paleta actual;

un elemento sintáctico que indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape;

30 un elemento sintáctico que indica un número de índices de la paleta actual que se señalan o deducen explícitamente; y

35 uno o más elementos sintácticos que indican unos índices de una matriz de entradas de paleta actual; y

descodificar el bloque actual de datos de vídeo en base a la paleta para el bloque actual de datos de vídeo, el grupo de elementos sintácticos y el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo.

40 2. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:

45 codificar, en una corriente de bits de vídeo codificada y usando codificación aritmética binaria adaptativa a contexto (CABAC) con un contexto, un elemento sintáctico, `palette_transpose_flag`, que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para un bloque actual de datos de vídeo;

50 codificar, en la corriente de bits de vídeo codificada, usando CABAC con un contexto y en una posición en la corriente de bits de vídeo codificada que está directamente después de la `palette_transpose_flag`, uno o más elementos sintácticos relacionados con un parámetro de cuantificación (QP) delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo;

55 codificar, en la corriente de bits de vídeo codificada, un grupo de elementos sintácticos consecutivos usando un modo de derivación, en el que el grupo comprende:

uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si unas entradas de una paleta predictora se reúsan en la paleta actual;

60 un elemento sintáctico que indica un número de entradas de la paleta actual que se señalan explícitamente;

uno o más elementos sintácticos que indican cada uno un valor de un componente de una entrada de la paleta actual;

65 un elemento sintáctico que indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape;



- un elemento sintáctico que indica un número de índices de la paleta actual que se señalan o deducen explícitamente; y
- 5 uno o más elementos sintácticos que indican unos índices de una matriz de entradas de paleta actual; y
- codificar el bloque actual de datos de vídeo en base a la paleta para el bloque actual de datos de vídeo, el grupo de elementos sintácticos y el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo.
- 10 3. El procedimiento de la reivindicación 1, en el que el elemento sintáctico que indica si el proceso de transposición se aplica a unos índices de paleta del bloque actual de datos de vídeo comprende un elemento sintáctico `palette_transpose_flag`.
- 15 4. El procedimiento de la reivindicación 1, en el que el uno o más elementos sintácticos relacionados con un QP delta comprenden uno o ambos de un elemento sintáctico que indica un valor absoluto de una diferencia entre un QP del bloque actual y un predictor del QP del bloque actual y un elemento sintáctico que indica un signo de la diferencia entre el QP del bloque actual y el predictor del QP del bloque actual.
- 20 5. El procedimiento de la reivindicación 1, en el que el uno o más elementos sintácticos relacionados con unos desplazamientos de QP de croma comprenden uno o ambos de un elemento sintáctico que indica si unas entradas de una o más listas de desplazamiento se añaden a un QP de luma del bloque actual para determinar unos QP de croma para el bloque actual y un elemento sintáctico que indica un índice de una entrada en cada una de la una o más listas de desplazamiento que se añaden al QP de luma para el bloque actual para determinar los QP de croma para el bloque actual.
- 25 6. El procedimiento de la reivindicación 1, en el que uno o más de:
- 30 el uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si unas entradas de una paleta predictora se reúsan en la paleta actual comprenden uno o más elementos sintácticos `palette_predictor_run`,
- el elemento sintáctico que indica un número de entradas de la paleta actual que se señalan explícitamente comprende un elemento sintáctico `num_signalled_palette_entries`,
- 35 el uno o más elementos sintácticos que indican cada uno un valor de un componente de una entrada de la paleta actual comprenden uno o más elementos sintácticos `palette_entry`,
- el elemento sintáctico que indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape comprende `palette_escape_val_present_flag`,
- 40 el elemento sintáctico que indica un número de índices de la paleta actual que se señalan o deducen explícitamente comprende un elemento sintáctico `num_palette_indices_idc`, y
- 45 el uno o más elementos sintácticos que indican índices de una matriz de entradas de paleta actual comprenden uno o más elementos sintácticos `palette_index_idc`.
- 50 7. El procedimiento de la reivindicación 1, en el que descodificar el grupo de elementos sintácticos comprende descodificar el grupo de elementos sintácticos a partir de la corriente de bits de vídeo codificada en una posición de la corriente de bits de vídeo codificada que está antes del elemento sintáctico que indica si el proceso de transposición se aplica a unos índices de paleta del bloque actual de datos de vídeo.
8. El procedimiento de la reivindicación 1, que comprende además:
- 55 descodificar, a partir de la corriente de bits de vídeo codificada después del grupo de elementos sintácticos codificados usando un modo de derivación, un elemento sintáctico que indica una última aparición de una marca de tipo de recorrido dentro del bloque actual de datos de vídeo.
- 60 9. El procedimiento de la reivindicación 9, en el que descodificar el elemento sintáctico que indica la última aparición de una marca de tipo de recorrido dentro del bloque actual de datos de vídeo comprende descodificar el elemento sintáctico que indica la última aparición de una marca de tipo de recorrido dentro del bloque actual de datos de vídeo usando una codificación aritmética binaria adaptativa a contexto (CABAC) con un contexto.
- 65 10. Un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo:
- una memoria configurada para almacenar datos de vídeo; y

uno o más procesadores configurados para llevar a cabo las etapas de la reivindicación 2.

- 5 11. Un dispositivo para descodificar datos de vídeo, comprendiendo el dispositivo:

una memoria configurada para almacenar datos de vídeo; y

uno o más procesadores configurados para llevar a cabo las etapas de una cualquiera de las reivindicaciones 1, 3-9.

- 10 12. Un dispositivo para descodificar datos de vídeo, comprendiendo el dispositivo:

medios para descodificar, a partir de una corriente de bits de vídeo codificada y usando una codificación aritmética binaria adaptativa a contexto (CABAC) con un contexto, un elemento sintáctico, `palette_transpose_flag`, que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para un bloque actual de datos de vídeo;

medios para descodificar, a partir de la corriente de bits de vídeo codificada, usando CABAC con un contexto y en una posición en la corriente de bits de vídeo codificada que está directamente después de la `palette_transpose_flag`, uno o más elementos sintácticos relacionados con un parámetro de cuantificación (QP) delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo;

medios para descodificar, a partir de la corriente de bits de vídeo codificada, un grupo de elementos sintácticos consecutivos usando un modo de derivación, en el que el grupo comprende:

25 uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si unas entradas de una paleta predictora se reúsan en la paleta actual;

30 un elemento sintáctico que indica un número de entradas de la paleta actual que se señalan explícitamente;

uno o más elementos sintácticos que indican cada uno un valor de un componente de una entrada de la paleta actual;

35 un elemento sintáctico que indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape;

40 un elemento sintáctico que indica un número de índices de la paleta actual que se señalan o deducen explícitamente; y

uno o más elementos sintácticos que indican unos índices de una matriz de entradas de paleta actual; y

45 medios para descodificar el bloque actual de datos de vídeo en base a la paleta para el bloque actual de datos de vídeo, el grupo de elementos sintácticos y el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo.

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

50 medios para codificar, en una corriente de bits de vídeo codificada y usando codificación aritmética binaria adaptativa a contexto (CABAC) con un contexto, un elemento sintáctico, `palette_transpose_flag`, que indica si un proceso de transposición se aplica a unos índices de paleta de una paleta para un bloque actual de datos de vídeo;

55 medios para codificar, en la corriente de bits de vídeo codificada, usando CABAC con un contexto y en una posición de la corriente de bits de vídeo codificada que está directamente después de la `palette_transpose_flag`, uno o más elementos sintácticos relacionados con un parámetro de cuantificación (QP) delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo;

60 medios para codificar, en la corriente de bits de vídeo codificada, un grupo de elementos sintácticos consecutivos usando un modo de derivación, en el que el grupo comprende:

uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si unas entradas de una paleta predictora se reúsan en la paleta actual;

65 un elemento sintáctico que indica un número de entradas de la paleta actual que se señalan explícitamente;

- uno o más elementos sintácticos que indican cada uno un valor de un componente de una entrada de la paleta actual;
- 5 un elemento sintáctico que indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape;
- un elemento sintáctico que indica un número de índices de la paleta actual que se señalan o deducen explícitamente; y
- 10 uno o más elementos sintácticos que indican unos índices de una matriz de entradas de paleta actual; y
- medios para codificar el bloque actual de datos de vídeo en base a la paleta para el bloque actual de datos de vídeo, el grupo de elementos sintácticos y el uno o más elementos sintácticos relacionados con un QP delta y/o unos desplazamientos de QP de croma para el bloque actual de datos de vídeo.
- 15
- 14.** Un medio de almacenamiento legible por ordenador que almacena al menos una parte de una corriente de bits de vídeo codificada que, cuando se procesa mediante un dispositivo de descodificación de vídeo, hace que uno o más procesadores del dispositivo de descodificación de vídeo:
- 20 determinen si un proceso de transposición se aplica a unos índices de paleta de una paleta para un bloque actual de datos de vídeo;
- generen la paleta para el bloque actual de datos de vídeo en base a un siguiente grupo de elementos sintácticos consecutivos en la parte de la corriente de bits de vídeo codificada:
- 25 uno o más elementos sintácticos que indican un número de ceros que preceden a una entrada no cero en una matriz que indica si unas entradas de una paleta predictora se reúsan en la paleta actual;
- 30 un elemento sintáctico que indica un número de entradas de la paleta actual que se señalan explícitamente;
- uno o más elementos sintácticos que indican cada uno un valor de un componente de una entrada de la paleta actual;
- 35 un elemento sintáctico que indica si el bloque actual de datos de vídeo incluye al menos una muestra codificada de escape;
- un elemento sintáctico que indica un número de índices de la paleta actual que se señalan o deducen explícitamente; y
- 40 uno o más elementos sintácticos que indican unos índices de una matriz de entradas de paleta actual; y
- descodifiquen el bloque actual de los datos de vídeo en base a la paleta para el bloque actual de datos de vídeo y un parámetro de cuantificación (QP) delta y uno o más desplazamientos de QP de croma para el bloque actual de datos de vídeo,
- 45 en el que uno o más elementos sintácticos relacionados con el QP delta y uno o más elementos sintácticos relacionados con el uno o más desplazamientos de QP de croma para el bloque actual de datos de vídeo están localizados en una posición en la parte de la corriente de bits de vídeo codificada que está directamente después de un elemento sintáctico, `palette_transpose_flag`, que indica si el proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo, en el que el elemento sintáctico que indica si el proceso de transposición se aplica a unos índices de paleta de la paleta para el bloque actual de datos de vídeo se descodifica usando codificación aritmética binaria adaptativa a contexto (CABAC) con un contexto, en el que al menos uno del uno o más elementos sintácticos relacionados con el QP delta y uno o más elementos sintácticos relacionados con el uno o más desplazamientos de QP de croma se descodifican usando CABAC con un contexto, y en el que el grupo de elementos sintácticos consecutivos se descodifican usando un modo de derivación.
- 50
- 55

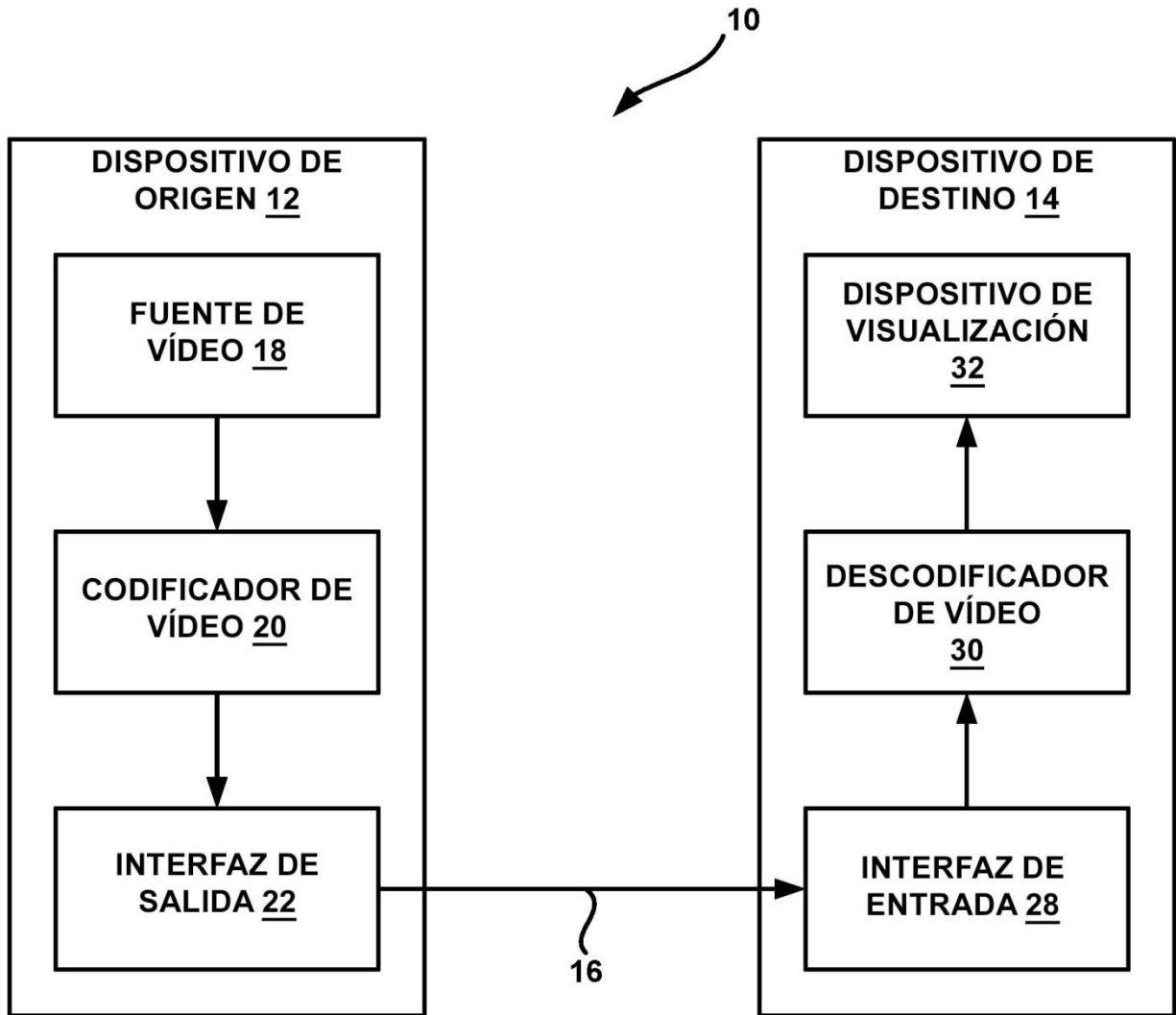


FIG. 1

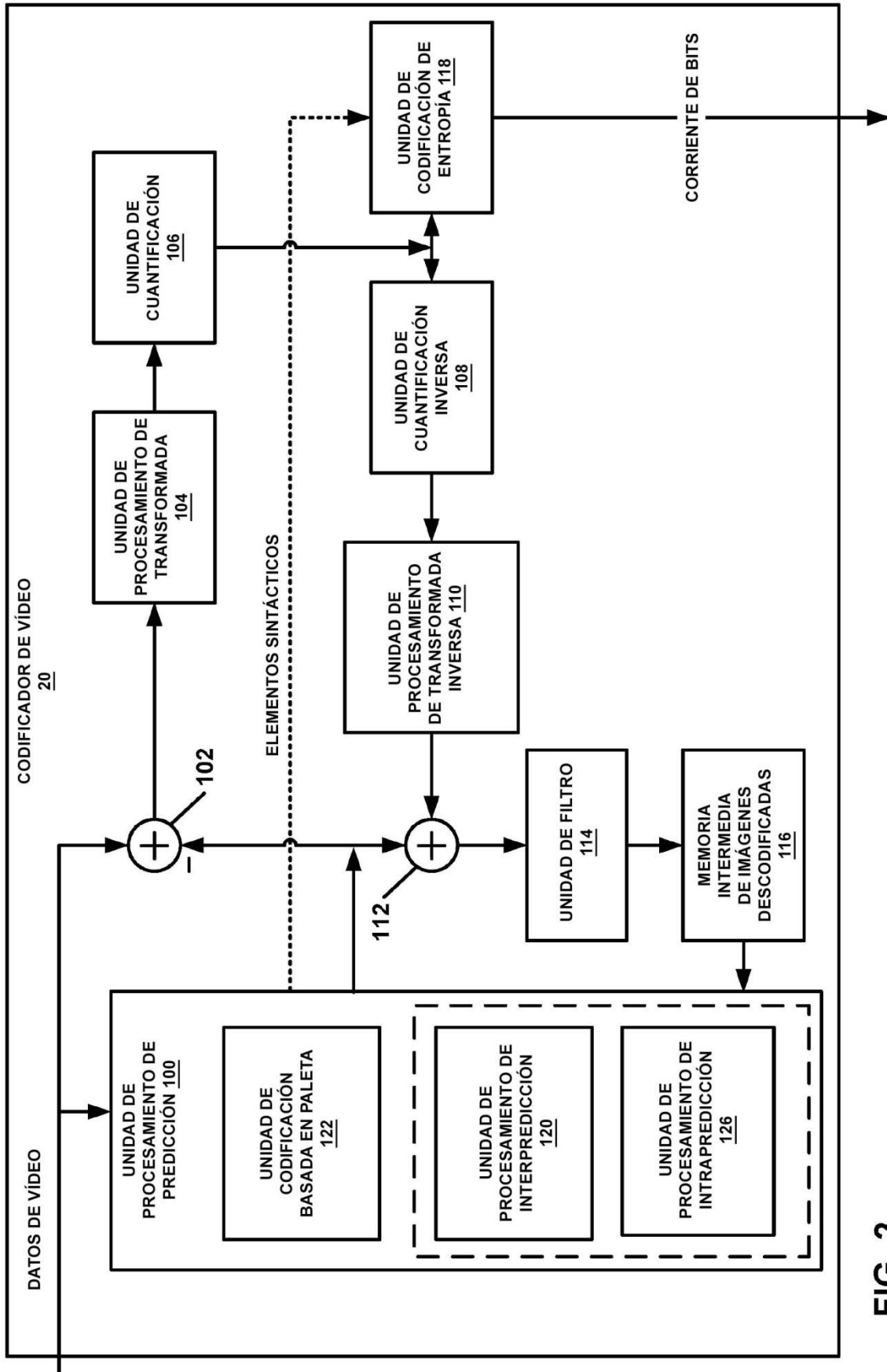


FIG. 2

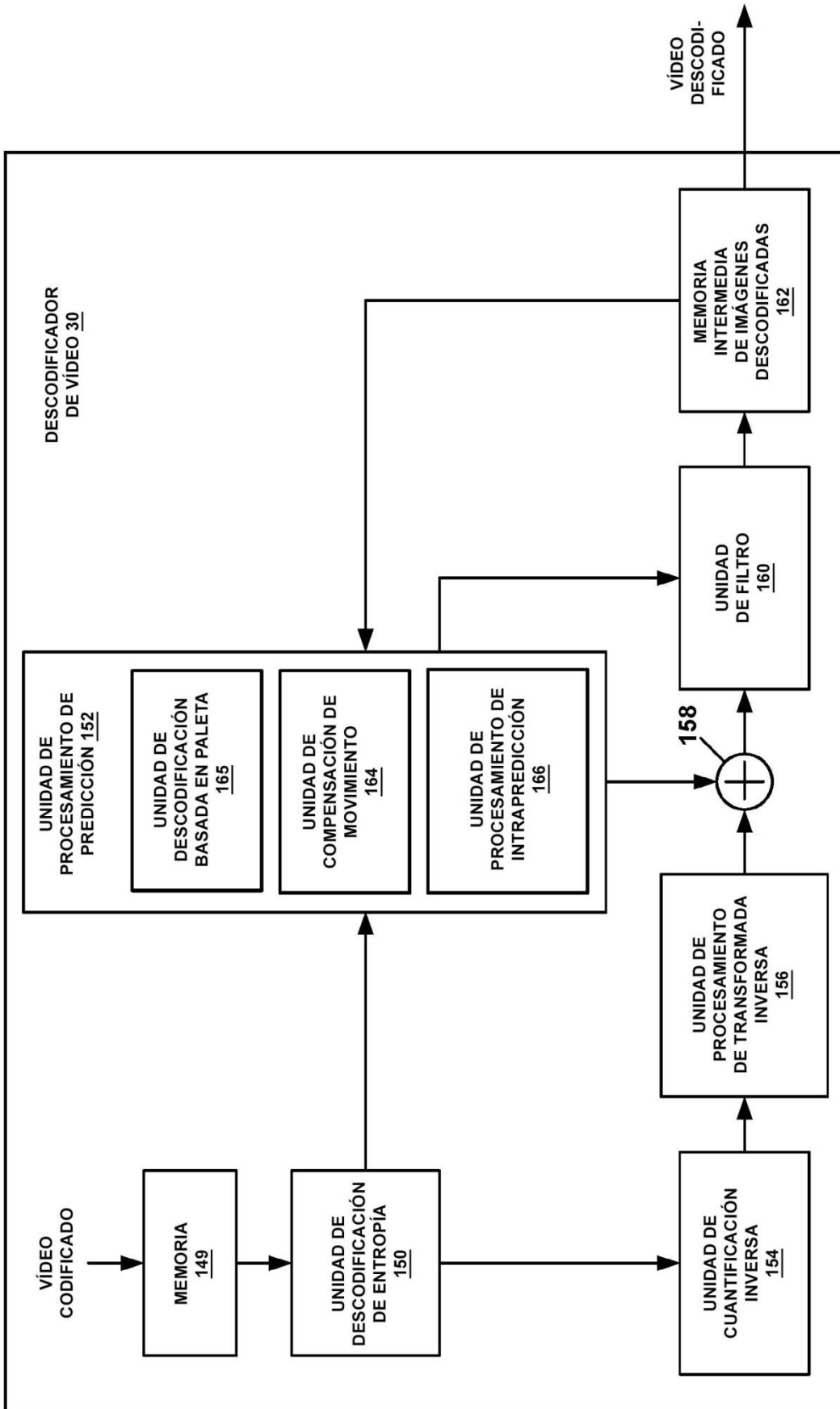


FIG. 3

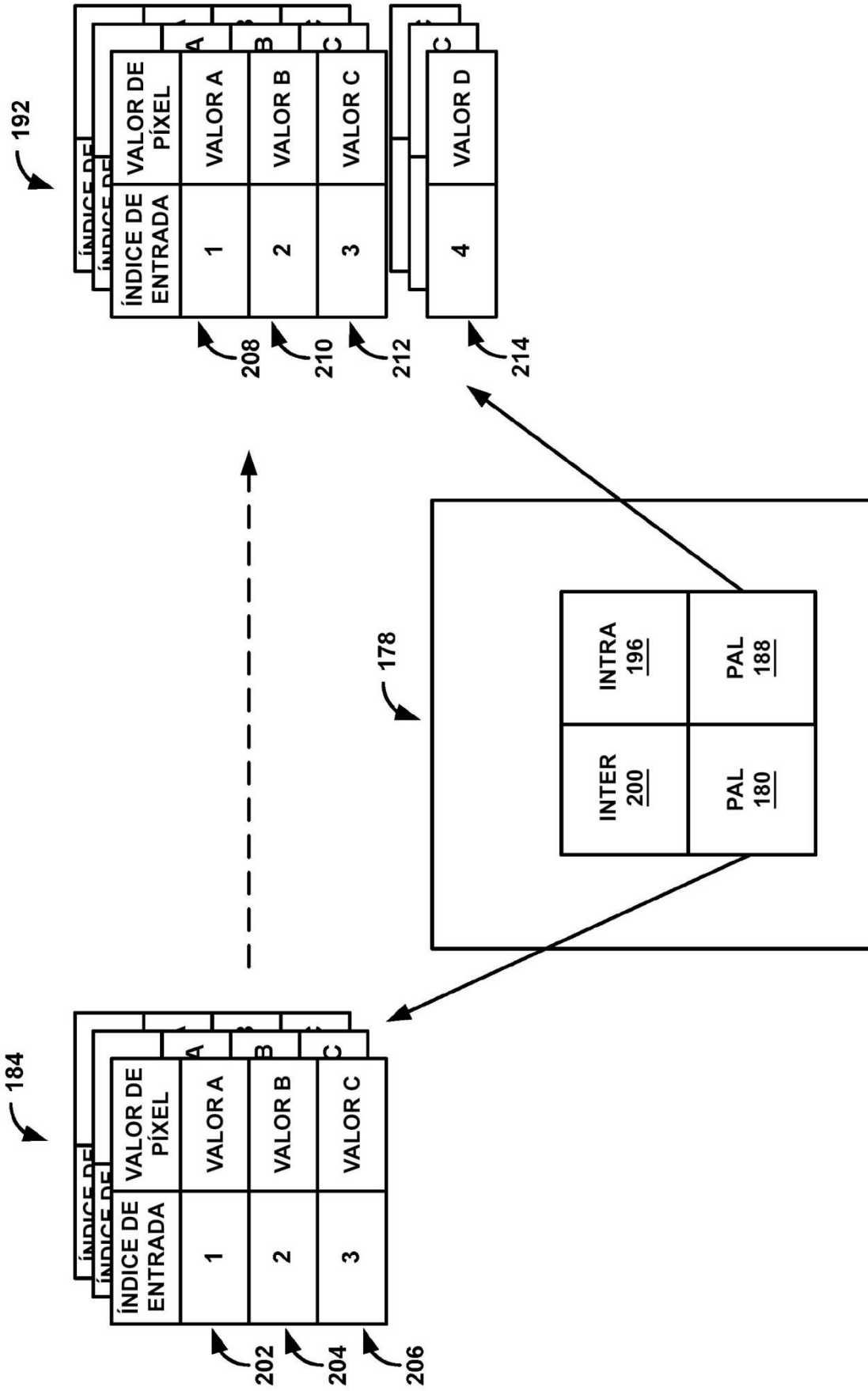


FIG. 4

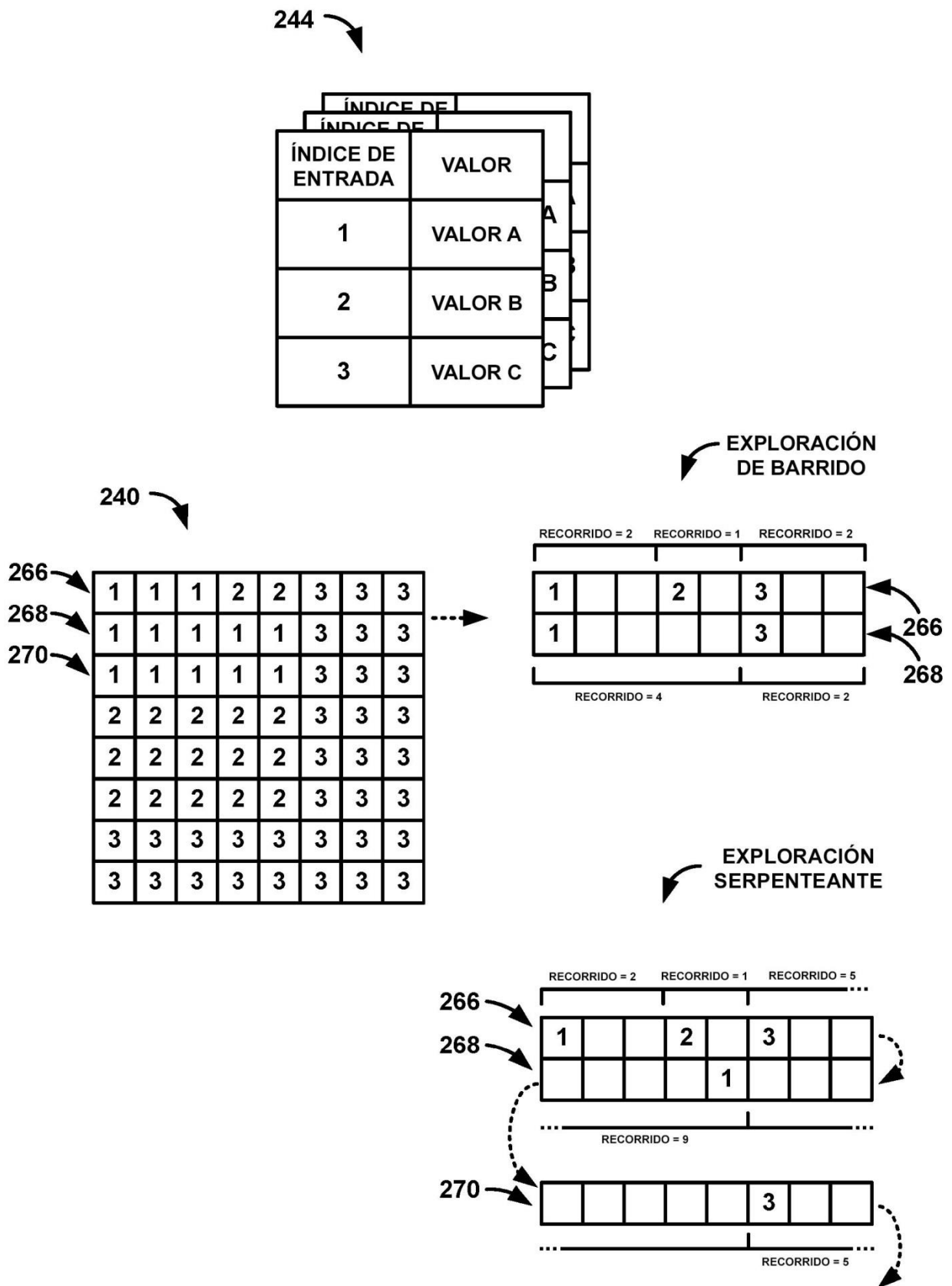


FIG. 5



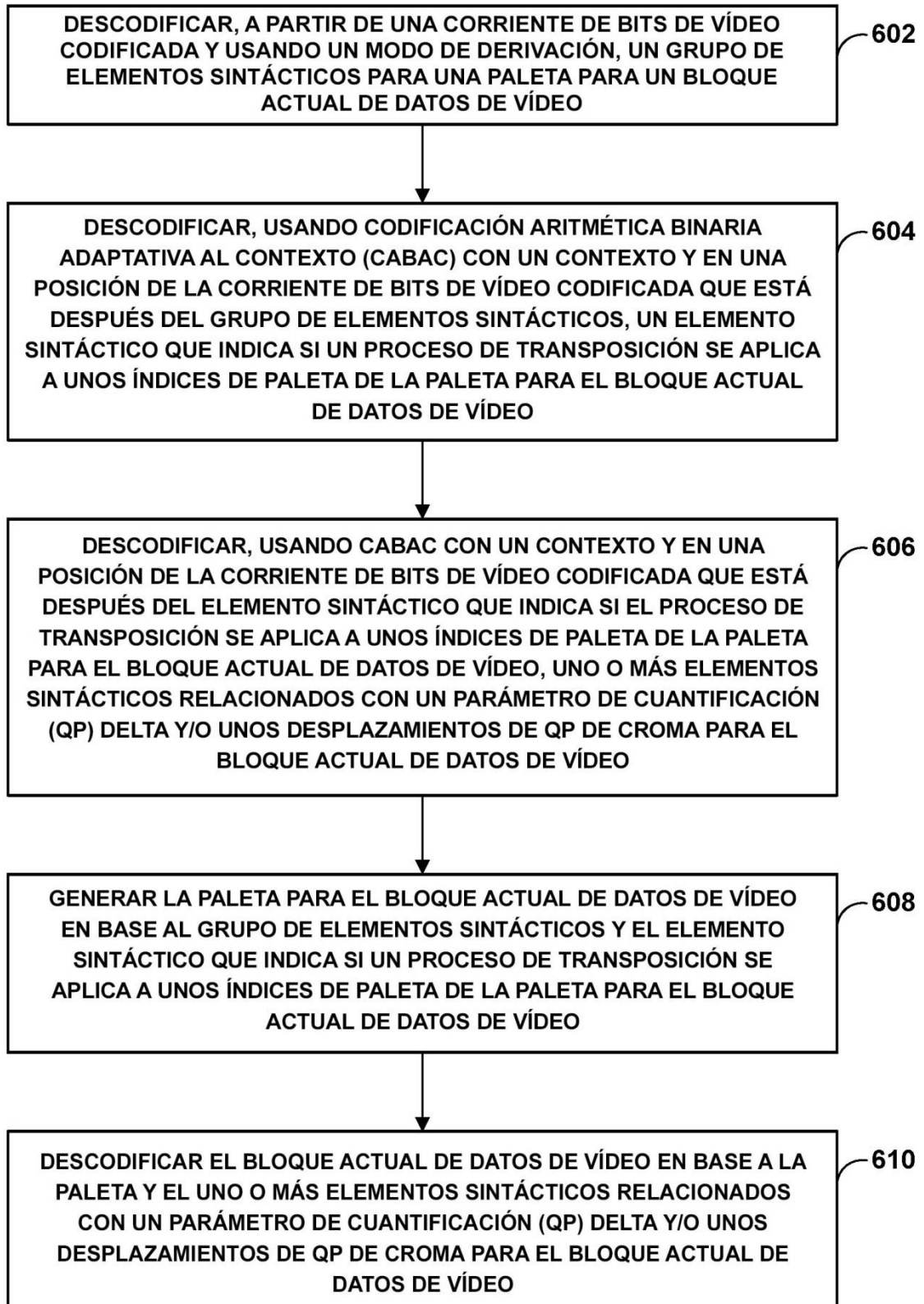
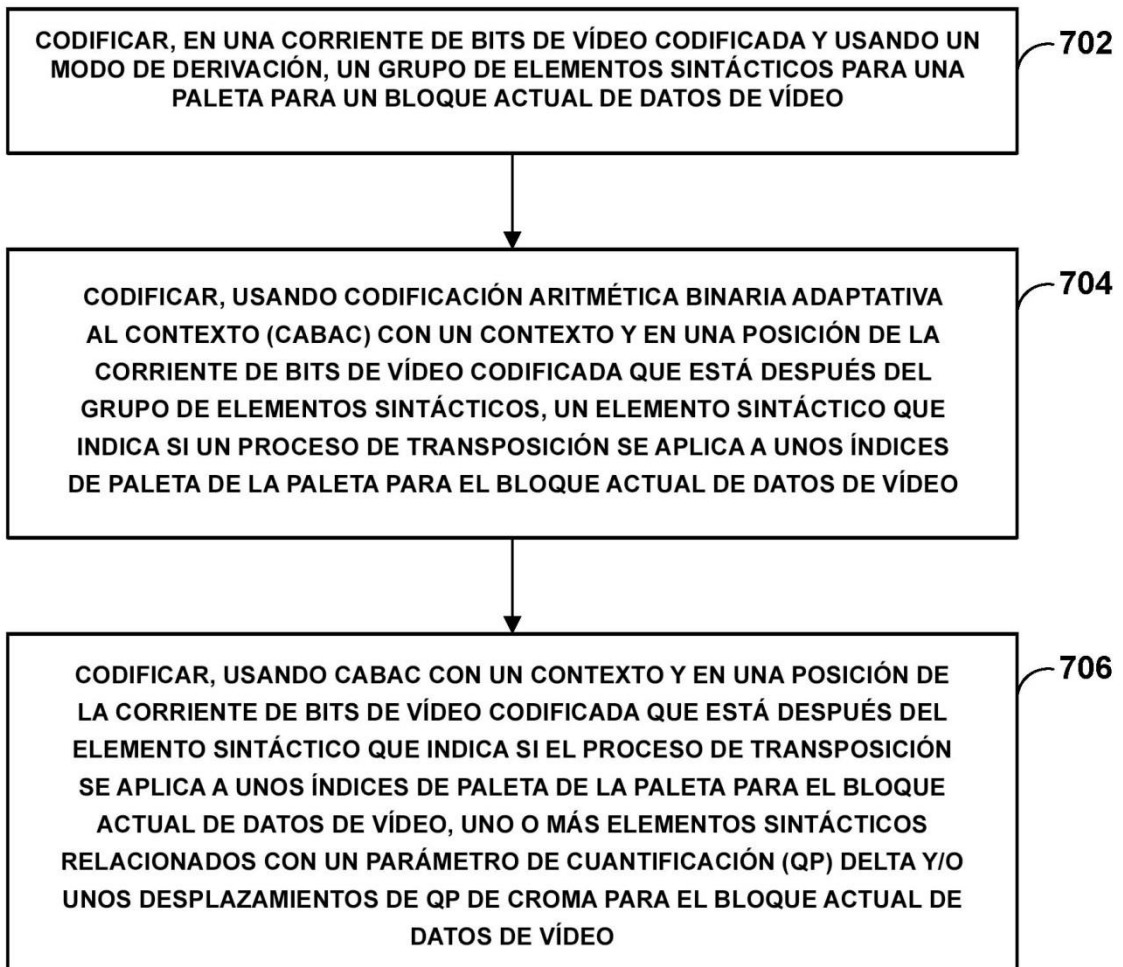


FIG. 6



**FIG. 7**