REIVINDICACIONES

1. Un método de decodificación de datos de video, el método comprende:

la decodificación, a partir de una corriente de bits de video codificado y el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto, de un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video;

la decodificación, a partir de la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video y el uso de CABAC con un contexto, de uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video;

la decodificación, a partir de la corriente de bits de video codificado, un grupo de elementos de sintaxis por el uso del modo de derivación, en el que el grupo comprende:

uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual;

un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita;

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

un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada;

un elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita; y

uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales; y

la decodificación del bloque actual de datos de video basado en paletas para el bloque actual de datos de video, el grupo de elementos de sintaxis y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

2. El método de acuerdo con la reivindicación 1, en el que el elemento de sintaxis

que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video comprende un elemento de sintaxis palette_transpose_flag.

3. El método de acuerdo con la reivindicación 1, en el que los uno o más elementos de sintaxis relacionados con QP delta comprenden uno o ambos de un elemento de sintaxis 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 de sintaxis que indica un signo de la diferencia entre el QP del bloque actual y el predictor del QP del bloque actual.

4. El método de acuerdo con la reivindicación 1, en el que los uno o más elementos de sintaxis relacionados con compensaciones de QP croma comprenden uno o ambos de un elemento de sintaxis que indica si las entradas en una o más listas de compensaciones se añaden a un QP luma del bloque actual para determinar los QP croma para el bloque actual y un elemento de sintaxis que indica un índice de una entrada en cada una de las una o más listas de compensaciones que se agregan al QP luma para el bloque actual para determinar los QP croma para el bloque actual.

5. El método de acuerdo con la reivindicación 1, en el que uno o más de:

los uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual comprenden uno o más elementos de sintaxis palette_predictor_run,

el elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita comprende un elemento de sintaxis num_signalled_palette_entries,

los uno o más elementos de sintaxis que indican cada uno un valor de un componente en una entrada en la paleta actual comprenden uno o más elementos de sintaxis palette_entry,

el elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos un muestra de escape codificada comprende palette_escape_val_present_flag,

el elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita comprende un elemento de sintaxis num_palette_indices_idc, y

los uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales comprenden uno o más elementos de sintaxis palette_index_idc.

6. El método de acuerdo con la reivindicación 1, en el que la decodificación del grupo

de elementos de sintaxis comprende la decodificación del grupo de elementos de sintaxis de la corriente de bits de video codificado en una posición en la corriente de bits de video codificado que se encuentra antes del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video.

7. El método de acuerdo con la reivindicación 1, que además comprende:

la decodificación, a partir de la corriente de bits de video codificado después de que el grupo de elementos de sintaxis se codifica por el uso del modo de derivación, un elemento de sintaxis que indica una última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video.

8. El método de acuerdo con la reivindicación 7, en el que la decodificación del elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video comprende la decodificación del elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video por el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto.

9. Un método de codificación de datos de video, el método comprende:

la codificación, en una corriente de bits de video codificado y el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto, de un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video;

la codificación, en la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video y el uso de CABAC con un contexto, de uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video;

la codificación, a partir de la corriente de bits de video codificado, un grupo de elementos de sintaxis por el uso del modo de derivación, en el que el grupo comprende:

uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual;

un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita;

uno o más elementos de sintaxis que indican cada uno un valor de un

componente en una entrada en la paleta actual;

un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada;

un elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita; y

uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales; y

la codificación del bloque actual de datos de video basado en paletas para el bloque actual de datos de video, el grupo de elementos de sintaxis y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

10. El método de acuerdo con la reivindicación 9, en el que el elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video comprende un elemento de sintaxis palette_transpose_flag.

11. El método de acuerdo con la reivindicación 9, en el que los uno o más elementos de sintaxis relacionados con QP delta comprenden uno o ambos de un elemento de sintaxis 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 de sintaxis que indica un signo de la diferencia entre el QP del bloque actual y el predictor del QP del bloque actual.

12. El método de acuerdo con la reivindicación 9, en el que los uno o más elementos de sintaxis relacionados con compensaciones de QP croma comprenden uno o ambos de un elemento de sintaxis que indica si las entradas en una o más listas de compensaciones se añaden a un QP luma del bloque actual para determinar los QP croma para el bloque actual y un elemento de sintaxis que indica un índice de una entrada en cada una de las una o más listas de compensaciones que se agregan al QP luma para el bloque actual para determinar los QP croma para el bloque actual.

13. El método de acuerdo con la reivindicación 9, en el que uno o más de:

los uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual comprenden uno o más elementos de sintaxis, palette_predictor_run,

el elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita comprende un elemento de sintaxis num_signalled_palette_entries,

los uno o más elementos de sintaxis que indican cada uno un valor de un componente en una entrada en la paleta actual comprenden uno o más elementos de sintaxis palette_entry,

el elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada comprende palette_escape_val_present_flag,

el elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita comprenden un elemento de sintaxis num_palette_indices_minus1, y

los uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales comprenden uno o más elementos de sintaxis palette_index_idc.

14. El método de acuerdo con la reivindicación 9, en el que la codificación del grupo de elementos de sintaxis comprende la codificación del grupo de elementos de sintaxis en la corriente de bits de video codificado en una posición en la corriente de bits de video codificado que se encuentra antes del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video.

15. El método de acuerdo con la reivindicación 9, que además comprende:

la codificación, en la corriente de bits de video codificado después de que el grupo de elementos de sintaxis se codifica por el uso del modo de derivación, de un elemento de sintaxis que indica una última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video.

16. El método de acuerdo con la reivindicación 15, en el que la codificación del elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video comprende la codificación del elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video por el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto.

17. Un dispositivo para la codificación o decodificación de datos de video, el dispositivo comprende:

una memoria configurada para almacenar datos de video;

uno o más procesadores configurados para:

codificar o decodificar, en una corriente de bits de video codificado y el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto, un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de

paleta de una paleta para un bloque actual de datos de video;

codificar o decodificar, en la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video y el uso de CABAC con un contexto, uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video;

codificar o decodificar, a partir de la corriente de bits de video codificado, un grupo de elementos de sintaxis por el uso del modo de derivación, en el que el grupo comprende:

uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual;

un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita;

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

un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada;

un elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita; y

uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales; y

codificar o decodificar el bloque actual de datos de video basado en paleta para el bloque actual de datos de video, el grupo de elementos de sintaxis y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

18. El dispositivo de acuerdo con la reivindicación 17, en el que el elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video comprende un elemento de sintaxis palette_transpose_flag.

19. El dispositivo de acuerdo con la reivindicación 17, en el que uno o más de:

los uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual comprenden uno o más

elementos de sintaxis, palette_predictor_run,

el elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita comprende un elemento de sintaxis num_signalled_palette_entries,

los uno o más elementos de sintaxis que indican cada uno un valor de un componente en una entrada en la paleta actual comprenden uno o más elementos de sintaxis palette_entry,

el elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada que comprende palette_escape_val_present_flag,

el elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas o inferidas en forma explícita comprende un elemento de sintaxis num_palette_indices_minus1, y

los uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales comprenden uno o más elementos de sintaxis palette_index_idc.

20. El dispositivo de acuerdo con la reivindicación 17, en el que, para codificar o decodificar el grupo de elementos de sintaxis, los uno o más procesadores están configurados para codificar o decodificar el grupo de elementos de sintaxis en la corriente de bits de video codificado en una posición en la corriente de bits de video codificado que se encuentra antes del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video.

21. El dispositivo de acuerdo con la reivindicación 17, en el que los uno o más procesadores están configurados además para:

codificar o decodificar, en la corriente de bits de video codificado después de que el grupo de elementos de sintaxis se codifica por el uso del modo de derivación, un elemento de sintaxis que indica una última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video.

22. El dispositivo de acuerdo con la reivindicación 21, en el que, para codificar o decodificar el elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video, los uno o más procesadores están configurados para codificar o decodificar el elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video por el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un

contexto.

23. Un dispositivo para la decodificación de datos de video, el dispositivo comprende:

un medio para la decodificación, a partir de una corriente de bits de video codificado y el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto, de un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video;

un medio para la decodificación, a partir de la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video y el uso de CABAC con el contexto, de uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video;

un medio para la decodificación, a partir de la corriente de bits de video codificado, un grupo de elementos de sintaxis por el uso del modo de derivación, en el que el grupo comprende:

uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual;

un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita;

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

un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada;

un elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita; y

uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales; y

un medio para la decodificación del bloque actual de datos de video basado en paletas para el bloque actual de datos de video, el grupo de elementos de sintaxis y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

24. Un medio de almacenamiento legible por computadora que almacena por lo menos

una porción de una corriente de bits de video codificado que, cuando se procesa por medio de un dispositivo de decodificación de video, provoca que uno o más procesadores del dispositivo de decodificación de video:

determinen si un proceso de transposición se aplica a los índices de paleta de una paleta para un bloque actual de datos de video;

generen la paleta para el bloque actual de los datos de video basados en:

uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual;

un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita;

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

un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada;

un elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita; y

uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales; y

decodifiquen el bloque actual de los datos de video basado en paletas para el bloque actual de datos de video y un parámetro de cuantificación delta (QP) y una o más compensaciones de QP croma para el bloque actual de datos de video,

en el que uno o más elementos de sintaxis relacionados con el QP delta y uno o más elementos de sintaxis relacionados con las una o más compensaciones de QP croma para el bloque actual de datos de video se encuentran en una posición en la corriente de bits de video codificado que se encuentra después de un elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video, en el que el elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video se decodifica usando la codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto, en el que al menos uno de los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y uno o más elementos de sintaxis relacionados con compensaciones de QP croma se decodifican utilizando CABAC con un contexto, y en el que el grupo de elementos de sintaxis se decodifican utilizando un modo

de derivación.

AGRUPAMIENTO DE COMPARTIMIENTOS DE DERIVACIÓN POR PALETAS PARA LA CODIFICACIÓN DE VIDEO

[0001] Esta solicitud reivindica el beneficio de la solicitud provisional de los Estados Unidos Núm. 62/175.137 presentada el 12 de junio de 2015, todo el contenido de la cual se incorpora como referencia en la presente memoria.

CAMPO TÉCNICO

[0002] La revelación se refiere a la codificación y decodificación de video.

ANTECENDENTES

[0003] Las capacidades de video digitales se pueden incorporar en una amplia gama de dispositivos, que incluyen televisores digitales, sistemas de radiodifusión directa digitales, sistemas de transmisión inalámbricos, asistentes digitales personales (PDA), computadoras portátiles o de escritorio, computadoras de tipo tablet, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de juegos de video, consolas de videojuegos, teléfonos celulares de radio o satélite, los denominados "teléfonos inteligentes", dispositivos de teleconferencia de video, dispositivos de streaming de video, y similares. Los dispositivos de video digitales implementan técnicas de compresión de video, tales como los descriptos en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de Video Avanzada (AVC), ITU-TH.265, el estándar de Codificación de Video de Alta Eficiencia (HEVC), y la extensión de tales estándares. Los dispositivos de video pueden transmitir, recibir, codificar, decodificar, y/o almacenar información de video digital de manera más eficiente por medio de la aplicación de tales técnicas de compresión de video.

[0004] Las técnicas de compresión de video llevan a cabo predicción (intra-imagen) espacial y/o (inter-imágenes) la predicción temporal para reducir o eliminar la redundancia inherente a las secuencias de video. Para la codificación de video basada en bloques, un corte de video (es decir, una trama de video o una porción de una trama de video) se puede dividir en bloques de video. Los bloques de video en un corte (I) intra-codificado de una imagen se codifican por el uso de predicción espacial con respecto a las muestras de referencia en bloques vecinos de la misma imagen. Los bloques de video en un corte inter-codificado (P o B) de una imagen pueden utilizar predicción espacial con respecto a las muestras de referencia en bloques de vecinos de la misma imagen o predicción temporal con respecto a las muestras de

referencia en otras imágenes de referencia. Las imágenes se pueden denominar como tramas, imágenes de referencia y se pueden referir a tramas de referencia.

[0005] La predicción espacial o temporal da lugar a un bloque de predicción para un bloque que se han codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que ha de codificarse y el bloque de predicción. Un bloque inter-codificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestra de referencia que forman el bloque de predicción, y los datos residuales indican la diferencia entre el bloque codificado y el bloque de predicción. Un bloque intra-codificado se codifica de acuerdo con un modo de intra-codificación y los datos residuales. Para una compresión adicional, los datos residuales pueden ser transformados desde el dominio de píxeles a un dominio de transformada, lo que da lugar a coeficientes residuales, que luego pueden ser cuantificados. Los coeficientes cuantificados, dispuestos inicialmente en una matriz de dos dimensiones, pueden ser escaneados con el fin de producir un vector unidimensional de coeficientes, y se puede aplicar codificación de la entropía para lograr aún más compresión.

SÍNTESIS

[0006] En un ejemplo, un método de decodificación de datos de video incluye la decodificación, a partir de una corriente de bits de video codificado, de un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video; la decodificación, a partir de la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video, uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video; y la decodificación del bloque actual de datos de video con base en la paleta para el bloque actual de datos de video y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

[0007] En otro ejemplo, un método de codificación de datos de video incluye la codificación, en una corriente de bits de video codificado, de un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video; la codificación, en la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición

se aplica a índices de paleta de la paleta para el bloque actual de datos de video, uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video; y la codificación del bloque actual de datos de video basado en paletas para el bloque actual de datos de video y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

[0008] En otro ejemplo, un dispositivo para la codificación de datos de video incluye una memoria configurada para almacenar datos de video y uno o más procesadores. En este ejemplo, el uno o más procesadores están configurados para: codificar, en una corriente de bits de video codificado, un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video; codificar, en la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video, uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video; y codificar el bloque actual de datos de video basado en paletas para el bloque actual de datos de video y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

[0009] En otro ejemplo, un dispositivo para la codificación de datos de video incluye un medio para la codificación, en una corriente de bits de video codificado, de un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video; un medio para la codificación, en la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video, de uno o más elementos de sintaxis relacionados con QP delta y/o compensaciones de QP croma para el bloque actual de datos de video; y un medio para la codificación del bloque actual de datos de video basado en paletas para el bloque actual de datos de video y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

[0010] En otro ejemplo, un medio de almacenamiento legible por computadora almacena instrucciones que, cuando se ejecutan, provocan que uno o más

procesadores de un dispositivo de codificación de video: codifiquen, en una corriente de bits de video codificado, un elemento de sintaxis que indica si un proceso de transposición se aplica a los índices de paleta de una paleta para un bloque actual de datos de video; codifiquen, en la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video, uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video; y codifiquen el bloque actual de datos de video basado en paletas para el bloque actual de datos de video y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

[0011] En otro ejemplo, un medio de almacenamiento legible por computadora almacena por lo menos una porción de una corriente de bits de video codificado que, cuando se procesa por medio de un dispositivo de decodificación de video, provoca que uno o más procesadores del dispositivo de decodificación de video: determinen si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video; y decodifiquen el bloque actual de los datos de video basado en paletas para el bloque actual de datos de video y un QP delta y una o más compensaciones de QP croma para el bloque actual de datos de video, en el que uno o más elementos de sintaxis relacionados con el QP delta y uno o más elementos de sintaxis relacionados con las una o más compensaciones de QP croma para el bloque actual de datos de video se encuentran en una posición en la corriente de bits de video codificado que se encuentra después de un elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video.

[0012] Los detalles de uno o más ejemplos se exponen en las figuras adjuntas y en la descripción a continuación. Otras características, objetivos y ventajas serán evidentes a partir de la descripción y las figuras, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LAS FIGURAS

[0013] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de video representativo que puede utilizar las técnicas descriptas en esta revelación.

[0014] La FIG. 2 es un diagrama de bloques que ilustra un codificador de video representativo que puede implementar las técnicas descriptas en esta revelación.

[0015] La FIG. 3 es un diagrama de bloques que ilustra un decodificador de video

representativo que puede implementar las técnicas descriptas en esta revelación.

[0016] La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de la determinación de una paleta para la codificación de datos de video, de acuerdo con las técnicas de esta revelación.

[0017] La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de la determinación de los índices para una paleta para un bloque de píxeles, de acuerdo con las técnicas de esta revelación.

[0018] La FIG. 6 es un diagrama de flujo que ilustra un proceso representativo para la decodificación de un bloque de datos de video por el uso del modo de paleta, de acuerdo con una o más técnicas de esta revelación.

[0019] La FIG. 7 es un diagrama de flujo que ilustra un proceso representativo para la codificación de un bloque de datos de video por el uso del modo de paleta, de acuerdo con una o más técnicas de esta revelación.

DESCRIPCIÓN DETALLADA

[0020] Esta revelación describe técnicas para la codificación y compresión de video. En particular, esta revelación describe técnicas para la codificación de datos de video basada en paletas. Por ejemplo, esta revelación describe técnicas para soportar la codificación del contenido de video, en especial contenido de pantalla con codificación por paletas, tal como técnicas para mejorar la binarización de índice de paleta, y las técnicas para la señalización para la codificación por paletas.

[0021] En la codificación de video tradicional, se supone que las imágenes son de tono continuo y espacialmente suaves. Partiendo de estas premisas, se han desarrollado varias herramientas, tal como transformada basada en bloques, filtrado, etc., y tales herramientas han demostrado un buen rendimiento para videos de contenido natural.

[0022] Sin embargo, en aplicaciones como escritorio remoto, trabajo colaborativo y las pantallas inalámbricas, el contenido de pantalla generado por computadora puede ser el contenido dominante a ser comprimido. Este tipo de contenido tiende a tener tonos discretos y cuentan con líneas nítidas y límites de los objetos con alto contraste. La asunción de tonos continuos y suavidad ya no se puede aplicar y por lo tanto las técnicas tradicionales de codificación de video puede no ser una forma eficaz para comprimir.

[0023] Con base en las características de video de contenido de la pantalla, se introduce la codificación por paletas para mejorar la eficacia de la codificación de contenido de pantalla (SCC) de acuerdo con lo propuesto en Guo *et al.*, "Palette Mode

for Screen Content Coding”, Equipo de Colaboración Conjunta en materia de Codificación de Video (JCT-VC) de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 13va Reunión: Incheon, KR, 18-26 Abr. 2013, Documento: JCTVC-M0323, disponible en http://fenix.it - sudparis.eu/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0323-v3.zip, (en adelante "JCTVC-M0323"). En forma específica, la codificación por paletas presenta una tabla de búsqueda, es decir, una paleta de colores, para comprimir los valores de píxel repetitivos basados en el hecho de que en SCC, los colores dentro de una CU suelen concentrarse en unos pocos valores pico. Teniendo en cuenta una paleta para una CU específica, los píxeles dentro de la CU se asignan a los índices de paleta. En la segunda etapa, se propone un método de longitud de ejecución de copia efectiva desde la izquierda para comprimir eficazmente el patrón repetitivo del bloque de índices. En algunos ejemplos, el modo de codificación de índice de paleta se puede generalizar tanto a la copia desde izquierda como a la copia desde arriba con la codificación de longitud de ejecución. Se debe tener en cuenta que, en algunos ejemplos, ningún proceso de transformación puede ser invocada para la codificación por paletas para evitar el desenfoque de los bordes afilados que pueden tener un gran impacto negativo sobre la calidad visual de los contenidos de pantalla.

[0024] De acuerdo con lo discutido con anterioridad, esta revelación describe la codificación basada en paletas, que puede ser en particular adecuada para la codificación de contenido generado en pantalla. Por ejemplo, si se supone que un área particular de datos de video tiene un número relativamente pequeño de colores, un codificador de video (un codificador de video o un decodificador de video) puede codificar una denominada "paleta" como una tabla de colores para la representación de los datos de video del área particular (por ej., un bloque dado). Cada píxel puede estar asociado con una entrada en la paleta que representa el color del píxel. Por ejemplo, el codificador de video puede codificar un índice que asigna el valor de píxel al valor apropiado en la paleta.

[0025] En el ejemplo anterior, un codificador de video puede codificar un bloque de datos de video por medio de la determinación de una paleta para el bloque, la localización de una entrada en paletas para representar el valor de color de cada píxel, y la codificación de la paleta con valores de índice para los píxeles que mapean el valor de píxel a la paleta. Un decodificador de video puede obtener, a partir de una corriente de bits codificada, una paleta para un bloque, así como también los valores de índice para los píxeles del bloque. El decodificador de video puede mapear los

valores de índice de los píxeles a las entradas de paleta para reconstruir los valores de píxel luma y croma del bloque.

[0026] El ejemplo anterior está destinado a proporcionar una descripción general de la codificación basada en paletas. En diversos ejemplos, las técnicas descriptas en esta revelación pueden incluir técnicas para varias combinaciones de uno o más modos de codificación basados en paletas de señalización, paletas de transmisión, paletas de predicción, paletas de derivación, y mapas de codificación basados en paletas de transmisión y otros elementos de sintaxis. Dichas técnicas pueden mejorar la eficacia de la codificación de video, por ej., lo que requiere un menor número de bits para representar el contenido generado por pantalla.

[0027] Por ejemplo, de acuerdo con los aspectos de esta revelación, un codificador de video (un codificador de video o un decodificador de video) puede codificar uno o más elementos de sintaxis para cada bloque que se codifica por el uso de un modo de codificación por paletas. Por ejemplo, el codificador de video puede codificar un **palette_mode_flag** para indicar si un modo de codificación basado en paletas se va a utilizar para la codificación de un bloque particular. En este ejemplo, un codificador de video puede codificar un **palette_mode_flag** con un valor que es igual a uno para especificar que el bloque que está siendo codificado ("bloque actual") se codifica por el uso de un modo de paleta. En este caso, un decodificador de video puede obtener el **palette_mode_flag** de la corriente de bits codificada y aplicar el modo de codificación basado en paletas para decodificar el bloque. En los casos en los que hay más de un modo de codificación basado en paletas disponible (por ej., hay más de una técnica basada en paletas disponible para la codificación), uno o más elementos de sintaxis pueden indicar uno de una pluralidad de diferentes modos de paleta para el bloque.

[0028] En algunos casos, el codificador de video puede codificar un **palette_mode_flag** con un valor que es igual a cero para especificar que la secuencia actual no está codificada por el uso de un modo de paleta. En tales casos, el codificador de video puede codificar el bloque por el uso de cualquiera de una variedad de modos de codificación de inter-predicción, de intra-predicción, o de otro tipo. Cuando el **palette_mode_flag** es igual a cero, el codificador de video puede codificar información adicional (por ej., elementos de sintaxis) para indicar el modo específico que se utiliza para codificar el bloque respectivo. En algunos ejemplos, de acuerdo con lo descripto a continuación, el modo puede ser un modo de codificación HEVC. El uso del **palette_mode_flag** se describe a modo de ejemplo. En otros ejemplos, otros elementos de sintaxis, tales como los códigos de múltiples bits se

pueden utilizar para indicar si el modo de codificación basado en paletas se ha de utilizar para uno o más bloques, o para indicar cuál de una pluralidad de modos se han de utilizar.

[0029] Cuando se utiliza un modo de codificación basado en paletas, una paleta puede ser transmitida por un codificador en la corriente de bits de datos de video codificada para su uso por parte de un decodificador. Una paleta puede ser transmitida para cada bloque o se puede compartir entre varios bloques en una imagen o un corte. La paleta se puede referir a una serie de valores de píxel que son dominantes y/o representativos del bloque, que incluyen, por ej., un valor de luma y dos valores de croma.

[0030] En algunos ejemplos, un elemento de sintaxis, tal como un marcador de transposición, se puede codificarse para indicar si un proceso de transposición se aplica a índices de paleta de una paleta actual. Si el marcador de transposición es cero, los índices de paleta para muestras pueden ser codificados en una exploración de desplazamiento horizontal. Del mismo modo, si el marcador de transposición es uno, los índices de paleta para muestras pueden ser codificados en una exploración de desplazamiento vertical. Esto puede ser pensado como la decodificación de los valores de índice suponiendo una exploración de desplazamiento horizontal y luego transponer el bloque (de filas a columnas).

[0031] Los aspectos de esta revelación incluyen técnicas para la codificación por paletas. Por ejemplo, de acuerdo con los aspectos de esta revelación, un codificador de video puede codificar uno o más elementos de sintaxis para definir una paleta. Algunos elementos de sintaxis representativos que un codificador de video puede codificar para definir una paleta actual para un bloque actual de datos de video incluyen, pero no se limitan a, un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de la paleta actual (por ej., **palette_transpose_flag**) (es decir, si el uno o más elementos de sintaxis relacionados con parámetro de cuantificación delta (QP) (por 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 de sintaxis relacionados con compensaciones de QP croma para el bloque actual de datos de video, uno o más elementos de sintaxis que indican un número de ceros que preceden una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual (por ej., **palette_predictor_run**), uno o más elementos de sintaxis que indican un número de entradas en la paleta actual que

están señalizadas en forma explícita (por ej., **num_signalled_palette_entries**), uno o más elementos de sintaxis que indican un valor de un componente en una entrada de paleta en la paleta actual (por ej., **palette_entry**), uno o más elementos de sintaxis que indican si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada (por ej., **palette_escape_val_present_flag**), uno o más elementos de sintaxis que indican un número de entradas en la paleta actual que están señalizadas o inferidas en forma explícita (por ej., **num_palette_indices_idc**), y uno o más elementos que indican los índices de una matriz de entradas de paleta actuales (por ej., **palette_index_idc**) sintaxis. Por ejemplo, cuando se opera de acuerdo con la Codificación de Contenido de Pantalla HEVC (SCC) Proyecto 3 (Joshi et al, "High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 3". Equipo de Colaboración Conjunta en materia de Codificación de Video (JCT-VC) del ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 20ma Reunión: Ginebra, CH, 10 de Febrero a 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, (de aquí en adelante "Proyecto 3 de HEVC SCC"), un codificador de video puede señalizar los elementos de sintaxis que figuran en la tabla de sintaxis palette_coding() (sección 7.3.8.8 del Proyecto 3 de HEVC SCC), se reproduce a continuación en el Tabla 1.

palette_coding(x0, y0, nCbS) {	**Descriptor**
palettePredictionFinished = 0	
NumPredictedPaletteEntries = 0	
para(i = 0; i < PredictorPaletteSize && !palettePredictionFinished && NumPredictedPaletteEntries< palette_max_size; i++) {	
palette_predictor_run	ue(v)
si(palette_predictor_run != 1) {	
si(palette_predictor_run > 1)	
i += palette_predictor_run − 1	
PalettePredictorEntryReuseFlag[i] = 1	
NumPredictedPaletteEntries++	
} si no	
palettePredictionFinished = 1	
}	
si(NumPredictedPaletteEntries < palette_max_size)	
num_signalled_palette_entries	ue(v)
numComps = (ChromaArrayType = = 0) ? 1 : 3	
para(cIdx = 0; cIdx < numComps; cIdx++)	

para(i = 0; i < num_signalled_palette_entries; i++)	
palette_entry	ae(v)
si(CurrentPaletteSize != 0)	
palette_escape_val_present_flag	ae(v)
si(palette_escape_val_present_flag) {	
si(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_palette_abs	ae(v)
si(cu_qp_delta_palette_abs)	
cu_qp_delta_palette_sign_flag	ae(v)
}	
si(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_palette_offset_flag	ae(v)
si(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
cu_chroma_qp_palette_offset_idx	ae(v)
}	
}	
si(MaxPaletteIndex > 0) {	
palette_transpose_flag	ae(v)
num_palette_indices_idc	ae(v)
para(i=0; i < NumPaletteIndices; i++) {	
palette_index_idc	ae(v)
PaletteIndexIdc[i] = palette_index_idc	
}	
last_palette_run_type_flag	ae(v)
}	
CurrNumIndices = 0	
PaletteScanPos = 0	
mientras que(PaletteScanPos <nCbS * nCbS) {	
xC = x0 + travScan[PaletteScanPos][0]	
yC = y0 + travScan[PaletteScanPos][1]	
si(PaletteScanPos > 0) {	
xcPrev = x0 + travScan[PaletteScanPos − 1][0]	
ycPrev = y0 + travScan[PaletteScanPos − 1][1]	
}	
PaletteRun = nCbS * nCbS − PaletteScanPos − 1	

si(MaxPaletteIndex > 0 && CurrNumIndices < NumPaletteIndices) {	
si(PaletteScanPos >= nCbS &&palette_run_type_flag[xcPrev][ycPrev] != COPY_ABOVE_MODE && PaletteScanPos < nCbS * nCbS – 1) {	
palette_run_type_flag[xC][yC]	ae(v)
}	
readIndex = 0	
si(palette_run_type_flag[xC][yC] = = COPY_INDEX_MODE && AdjustedMaxPaletteIndex > 0)	
readIndex = 1	
maxPaletteRun = nCbS * nCbS – PaletteScanPos – 1	
si(AdjustedMaxPaletteIndex > 0 && ((CurrNumIndices + readIndex) < NumPaletteIndices | | palette_run_type_flag[xC][yC] != last_palette_run_type_flag))	
si(maxPaletteRun > 0) {	
palette_run_msb_id_plus1	ae(v)
si(palette_run_msb_id_plus1 > 1)	
palette_run_refinement_bits	ae(v)
}	
CurrNumIndices + = readIndex	
}	
runPos = 0	
mientras que (runPos < = paletteRun) {	
xR = x0 + travScan[PaletteScanPos][0]	
yR = y0 + travScan[PaletteScanPos][1]	
si(palette_run_type_flag[xC][yC] = = COPY_INDEX_MODE) {	
PaletteSampleMode[xR][yR] = COPY_INDEX_MODE	
PaletteIndexMap[xR][yR] = CurrPaletteIndex	
} si no {	
PaletteSampleMode[xR][yR] = COPY_ABOVE_MODE	

PaletteIndexMap[xR][yR] = PaletteIndexMap[xR][yR − 1]	
}	
runPos++	
PaletteScanPos++	
}	
}	
si(palette_escape_val_present_flag) {	
sPos = 0	
mientras que(sPos <nCbS * nCbS) {	
xC = x0 + travScan[sPos][0]	
yC = y0 + travScan[sPos][1]	
si(PaletteIndexMap[xC][yC] = = MaxPaletteIndex) {	
para(cIdx = 0; cIdx < numComps; cIdx++)	
si(cIdx = = 0 | | (xR%2= =0 &&yR%2 = =0&& ChromaArrayType = = 1) | | (xR%2= = 0 && ChromaArrayType = = 2) | | ChromaArrayType = = 3) {	
palette_escape_val	ae(v)
PaletteEscapeVal[cIdx][xC][yC] = palette_escape_val	
}	
}	
sPos++	
}	
}	
}	

Tabla 1

[0032] Además de proporcionar un orden en el que los elementos de sintaxis se incluyen en una corriente de bits, la Tabla 1 también proporciona un descriptor para cada uno de los elementos de sintaxis que indica un tipo de codificación para cada elemento de sintaxis. Como un ejemplo, un codificador de video puede codificar elementos de sintaxis con el descriptor ue(v) por el uso de un número entero sin signo de 0-ésimo orden con códigos Exp-Golomb con el bit izquierdo primero. Como otro ejemplo, un codificador de video puede codificar elementos de sintaxis con el

descriptor ae(v) por el uso de códigos de entropía aritmética binaria adaptativa al contexto (CABAC). Cuando los compartimientos de un elemento de sintaxis se codifican por el uso de CABAC, un codificador de video puede codificar uno o más de los compartimientos por el uso de un contexto y/o puede codificar uno o más de los compartimientos sin un contexto. La codificación de un compartimiento por el uso de CABAC sin un contexto se puede denominar como el modo de derivación. El Proyecto 3 de HEVC SCC proporciona además una tabla (Tabla 9-47 del Proyecto 3 de HEVC SCC), que se reproduce parcialmente a continuación como la Tabla 2, que indica que los compartimientos de los elementos de sintaxis que figuran en la Tabla 1 codificados con contextos (es decir, de acuerdo con lo indicado por el contexto "0" y el contexto "1"), y dichos compartimientos están codificados en el modo de derivación.

Elemento de sintaxis	binIdx					
	0	1	2	3	4	>= 5
palette_predictor_run	derivar	Derivar	derivar	derivar	derivar	derivar
num_signalled_palette_entries	derivar	derivar	derivar	derivar	derivar	derivar
palette_entry	derivar	derivar	derivar	derivar	derivar	derivar
palette_escape_val_present_flag	derivar	na	na	na	na	na
cu_qp_delta_palette_abs	0	1	1	1	1	derivar
cu_qp_delta_palette_sign_flag	derivar	na	na	na	na	na
cu_chroma_qp_palette_offset_flag	0	na	na	na	na	na
cu_chroma_qp_palette_offset_idx	0	0	0	0	0	na
palette_transpose_flag	0	na	na	na	na	na
num_palette_indices_idc	derivar	derivar	derivar	derivar	derivar	derivar
last_palette_run_type_flag	0	na	na	na	na	na
palette_run_type_flag	0	na	Na	na	na	na
palette_index_idc	derivar	derivar	derivar	derivar	derivar	derivar
palette_run_msb_id_plus1	(cláusula 9.3.4.2.8)					
palette_run_refinement_bits	derivar	derivar	derivar	derivar	derivar	derivar
palette_escape_val	derivar	derivar	derivar	derivar	derivar	derivar

Tabla 2

[0033] Una comparación de la Tabla 1 y la Tabla 2 muestra que El Proyecto 3 de HEVC SCC prescribe que todos los elementos de sintaxis antes de **cu_qp_delta_palette_abs** (es decir, **num_signalled_palette_entries**, **palette_entry** y **palette_escape_val_present_flag**) están codificados por derivación. Del mismo modo, los elementos de sintaxis 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 por derivación.

[0034] Cuando se codifica un compartimiento por el uso de CABAC con un contexto, un codificador de video puede cargar el contexto del almacenamiento en la memoria. En algunos ejemplos, un codificador de video puede tener los recursos de memoria limitados disponibles y/o puede llevar mucho tiempo cargar un contexto en la memoria. Como tal, puede ser deseable para un codificador de video reducir al mínimo la cantidad de veces que los contextos se cargan en la memoria. En algunos ejemplos, el agrupamiento de compartimientos de derivación juntos puede reducir la cantidad de veces que los contextos se cargan en memoria, lo que puede aumentar el rendimiento de CABAC.

[0035] En YE *et al.*, "CE1-related: Palette Mode Context and Codeword Simplification", Equipo de Colaboración Conjunta en materia de Codificación de Video (JCT-VC) de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, Reunión 21: Varsovia, PL 19 a -26 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 (en adelante, "JCTVC-U0090"), se propuso que la **palette_transpose_flag** sea señalizada después de la **last_palette_run_type_flag**. En forma específica, JCTVC-U0090 propone la modificación de la tabla de sintaxis palette_coding() de acuerdo con lo mostrado a continuación en la Tabla 3 (donde el texto *en cursiva* se inserta y el texto en *[[cursiva con doble corchete]]* se elimina).

si(MaxPaletteIndex > 0) {	
[[palette_transpose_flag]]	*[[ae(v)]]*
num_palette_indices_idc	ae(v)
para(i=0; i < NumPaletteIndices; i++) {	
palette_index_idc	ae(v)
PaletteIndexIdc[i] = palette_index_idc	
}	
last_palette_run_type_flag	ae(v)
palette_transpose_flag	*ae(v)*
}	

Tabla 3

[0036] Sin embargo, en algunos ejemplos, la disposición de elementos de sintaxis propuestos por JCTVC-U0090 puede no ser óptima. Por ejemplo, cuando los elementos de sintaxis relacionados con QP delta (es decir, **cu_qp_delta_palette_abs** y **cu_qp_delta_palette_sign_flag**) y QP croma compensado (es decir,

cu_chroma_qp_palette_offset_flag y **cu_chroma_qp_palette_offset_idx**) están presentes, la disposición de los elementos de sintaxis propuestos por JCTVC-U0090 no puede resultar en el agrupamiento de compartimientos de derivación adicionales.

[0037] De acuerdo con una o más técnicas de esta revelación, un codificador de video puede codificar los elementos de sintaxis que se utilizan para definir una paleta actual de manera tal que los elementos de sintaxis que se codifican por el uso del modo de derivación sean codificados en forma consecutiva. Por ejemplo, en contraposición a la codificación de uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para un bloque actual de datos de video antes de que un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para el bloque actual de datos de video, un codificador de video puede codificar los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video después de que el elemento de sintaxis que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de video.

[0038] Un ejemplo de cómo la tabla de sintaxis palette_coding() se puede modificar para mover la señalización de los elementos de sintaxis relacionados con compensaciones QP delta y QP croma después de la palette_transpose_flag que se muestra a continuación en la Tabla 4 (donde el texto *en cursiva* se inserta y el texto en *[[cursiva con doble corchete]]* se elimina con relación a una versión anterior de la Tabla 4 en el Proyecto 3 de HEVC SCC).

palette_coding(x0, y0, nCbS) {	**Descriptor**
palettePredictionFinished = 0	
NumPredictedPaletteEntries = 0	
para(i = 0; i < PredictorPaletteSize && !palettePredictionFinished && NumPredictedPaletteEntries< palette_max_size; i++) {	
palette_predictor_run	ue(v)
si(palette_predictor_run != 1) {	
si(palette_predictor_run > 1)	
i += palette_predictor_run − 1	
PalettePredictorEntryReuseFlag[i] = 1	
NumPredictedPaletteEntries++	
} si no	
palettePredictionFinished = 1	

}	
si(NumPredictedPaletteEntries < palette_max_size)	
num_signalled_palette_entries	ue(v)
numComps = (ChromaArrayType = = 0) ? 1 : 3	
para(cIdx = 0; cIdx < numComps; cIdx++)	
para(i = 0; i < num_signalled_palette_entries; i++)	
palette_entry	ae(v)
si(CurrentPaletteSize != 0)	
palette_escape_val_present_flag	ae(v)
[[si(palette_escape_val_present_flag) {]]	
[[si(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {]]	
[[cu_qp_delta_palette_abs]]	*[[ae(v)]]*
[[si(cu_qp_delta_palette_abs)]]	
[[cu_qp_delta_palette_sign_flag]]	*[[ae(v)]]*
[[}]]	
[[si(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {]]	
[[cu_chroma_qp_palette_offset_flag]]	*[[ae(v)]]*
[[si(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)]]	
[[cu_chroma_qp_palette_offset_idx]]	*[[ae(v)]]*
[[}]]	
[[}]]	
si(MaxPaletteIndex > 0) {	
[[palette_transpose_flag]]	*[[ae(v)]]*
num_palette_indices_idc	ae(v)
para(i=0; i < NumPaletteIndices; i++) {	
palette_index_idc	ae(v)
PaletteIndexIdc[i] = palette_index_idc	
}	
last_palette_run_type_flag	ae(v)
palette_transpose_flag	*ae(v)*
}	
si(palette_escape_val_present_flag) {	

si(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_palette_abs	*ae(v)*
si(cu_qp_delta_palette_abs)	
cu_qp_delta_palette_sign_flag	*ae(v)*
}	
si(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_palette_offset_flag	*ae(v)*
si(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
cu_chroma_qp_palette_offset_idx	*ae(v)*
}	
}	
CurrNumIndices = 0	
PaletteScanPos = 0	
…	

Tabla 4

[0039] Por medio del movimiento de los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video después de que el elemento de sintaxis que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de video, el codificador de video puede agrupar (es decir, codificar en forma consecutiva) un mayor número de elementos de sintaxis que se codifican por el uso del modo de derivación. Por ejemplo, por medio del movimiento de los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video después de que el elemento de sintaxis que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de video, el codificador de video puede agrupar uno o más elementos de sintaxis que indican un número de entradas en la paleta actual que están señalizadas o inferidas en forma explícita (por ej., **num_palette_indices_idc**) y uno o más elementos de sintaxis con índices de entrada en una matriz de entradas de paleta actuales (por ej., **palette_index_idc**) con uno o más elementos de sintaxis relacionados con las compensaciones de QP croma para el bloque actual de datos de video, uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta

de predicción se vuelven a utilizar en la paleta actual (por ej., **palette_predictor_run**), uno o más elementos de sintaxis que indican un número de entradas en la paleta actual que están señalizadas en forma explícita (por ej., **num_signalled_palette_entries**), uno o más elementos de sintaxis que indican un valor de un componente en una entrada de paleta en la paleta actual (por ej., **palette_entry**), y uno o más elementos de sintaxis que indican si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada (por ej., **palette_escape_val_present_flag**). De esta manera, las técnicas de esta revelación pueden aumentar el rendimiento de CABAC, lo que puede reducir el tiempo necesario para codificar los datos de video por el uso del modo de codificación por paletas. Por ejemplo, por medio del agrupamiento de los elementos de sintaxis codificados por derivación, un codificador de video puede codificar secuencialmente los elementos de sintaxis agrupados sin iniciar, detener, reiniciar, volver a cargar, y restablecer un motor de codificación CABAC.

[0040] La Tabla 4 es sólo un ejemplo de cómo los elementos de sintaxis pueden estar dispuestos. En algunos ejemplos, los elementos de sintaxis relacionados con el desplazamiento QP delta y QP croma se puede mover más abajo en la tabla de sintaxis. Por ejemplo, los elementos de sintaxis relacionados con compensaciones de QP delta y QP croma se podrían colocar justo antes de los valores de los componentes de las muestras de escape (es decir, **palette_escape_val**). Un ejemplo de cómo los elementos de sintaxis relacionados con compensaciones de QP delta y QP croma se podrían colocar justo antes de los valores de los componentes para las muestras de escape se muestra a continuación en la Tabla 5 (donde el texto *en cursiva* se inserta y el texto en *[[cursiva con doble corchete]]* se elimina con relación al Proyecto 3 de HEVC SCC).

palette_coding(x0, y0, nCbS) {	**Descriptor**
palettePredictionFinished = 0	
NumPredictedPaletteEntries = 0	
para(i = 0; i < PredictorPaletteSize && !palettePredictionFinished && NumPredictedPaletteEntries< palette_max_size; i++) {	
palette_predictor_run	ue(v)
si(palette_predictor_run != 1) {	
si(palette_predictor_run > 1)	
i += palette_predictor_run – 1	
PalettePredictorEntryReuseFlag[i] = 1	

NumPredictedPaletteEntries++	
} si no	
palettePredictionFinished = 1	
}	
si(NumPredictedPaletteEntries < palette_max_size)	
num_signalled_palette_entries	ue(v)
numComps = (ChromaArrayType = = 0) ? 1 : 3	
para(cIdx = 0; cIdx < numComps; cIdx++)	
para(i = 0; i < num_signalled_palette_entries; i++)	
palette_entry	ae(v)
si(CurrentPaletteSize != 0)	
palette_escape_val_present_flag	ae(v)
[[si(palette_escape_val_present_flag) {]]	
[[si(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {]]	
[[cu_qp_delta_palette_abs]]	*[[ae(v)]]*
[[si(cu_qp_delta_palette_abs)]]	
[[cu_qp_delta_palette_sign_flag]]	*[[ae(v)]]*
[[}]]	
[[si(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {]]	
[[cu_chroma_qp_palette_offset_flag]]	*[[ae(v)]]*
[[si(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)]]	
[[cu_chroma_qp_palette_offset_idx]]	*[[ae(v)]]*
[[}]]	
[[}]]	
si(MaxPaletteIndex > 0) {	
~~***[[palette_transpose_flag]]***~~	~~*[[ae(v)]]*~~
num_palette_indices_idc	ae(v)
para(i=0; i < NumPaletteIndices; i++) {	
palette_index_idc	ae(v)
PaletteIndexIdc[i] = palette_index_idc	
}	
last_palette_run_type_flag	ae(v)

palette_transpose_flag	*ae(v)*
}	
CurrNumIndices = 0	
PaletteScanPos = 0	
mientras que(PaletteScanPos <nCbS * nCbS) {	
xC = x0 + travScan[PaletteScanPos][0]	
yC = y0 + travScan[PaletteScanPos][1]	
si(PaletteScanPos > 0) {	
xcPrev = x0 + travScan[PaletteScanPos − 1][0]	
ycPrev = y0 + travScan[PaletteScanPos − 1][1]	
}	
PaletteRun = nCbS * nCbS − PaletteScanPos − 1	
si(MaxPaletteIndex > 0 && CurrNumIndices < NumPaletteIndices) {	
si(PaletteScanPos >= nCbS &&palette_run_type_flag[xcPrev][ycPrev] != COPY_ABOVE_MODE && PaletteScanPos < nCbS * nCbS – 1) {	
palette_run_type_flag[xC][yC]	ae(v)
}	
readIndex = 0	
si(palette_run_type_flag[xC][yC] = = COPY_INDEX_MODE && AdjustedMaxPaletteIndex > 0)	
readIndex = 1	
maxPaletteRun = nCbS * nCbS – PaletteScanPos – 1	
si(AdjustedMaxPaletteIndex > 0 && ((CurrNumIndices + readIndex) < NumPaletteIndices | | palette_run_type_flag[xC][yC] != last_palette_run_type_flag))	
si(maxPaletteRun > 0) {	
palette_run_msb_id_plus1	ae(v)
si(palette_run_msb_id_plus1 > 1)	

palette_run_refinement_bits	ae(v)
}	
CurrNumIndices + = readIndex	
}	
runPos = 0	
mientras que (runPos < = paletteRun) {	
xR = x0 + travScan[PaletteScanPos][0]	
yR = y0 + travScan[PaletteScanPos][1]	
si(palette_run_type_flag[xC][yC] = = COPY_INDEX_MODE) {	
PaletteSampleMode[xR][yR] = COPY_INDEX_MODE	
PaletteIndexMap[xR][yR] = CurrPaletteIndex	
} si no {	
PaletteSampleMode[xR][yR] = COPY_ABOVE_MODE	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xR][yR − 1]	
}	
runPos++	
PaletteScanPos++	
}	
}	
si(palette_escape_val_present_flag) {	
si(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_palette_abs	*ae(v)*
si(cu_qp_delta_palette_abs)	
cu_qp_delta_palette_sign_flag	*ae(v)*
}	
si(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_palette_offset_flag	*ae(v)*
si(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	

cu_chroma_qp_palette_offset_idx	*ae(v)*
}	
sPos = 0	
mientras que(sPos <nCbS * nCbS) {	
xC = x0 + travScan[sPos][0]	
yC = y0 + travScan[sPos][1]	
si(PaletteIndexMap[xC][yC] = = MaxPaletteIndex) {	
para(cIdx = 0; cIdx < numComps; cIdx++)	
si(cIdx = = 0 | | (xR%2= =0 &&yR%2 = =0&& ChromaArrayType = = 1) | | (xR%2= = 0 && ChromaArrayType = = 2) | | ChromaArrayType = = 3) {	
palette_escape_val	ae(v)
PaletteEscapeVal[cIdx][xC][yC] = palette_escape_val	
}	
}	
sPos++	
}	
}	
}	

Tabla 5

[0041] Las técnicas para la codificación basada en paletas de datos de video se pueden utilizar con una o más otras técnicas de codificación, tales como las técnicas de codificación inter o intra-predictiva. Por ejemplo, de acuerdo con lo descripto en mayor detalle a continuación, un codificador o decodificador, o el codificador-decodificador combinado (códec), puede estar configurado para llevar a cabo la codificación de inter e intra-predicción, así como también la codificación basada en paletas.

[0042] En algunos ejemplos, las técnicas de codificación basada en paletas pueden estar configuradas para su uso con los estándares de codificación de uno o más de video. Algunos ejemplos de estándares video de codificación incluyen, pero no se limitan a, ITU-T H.261, ISO/IEC MPEG-1 ITU-T H.262 visual, o ISO/IEC MPEG-2

Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual y el ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), que incluyen sus extensiones Codificación de Video Escalable (SVC) y Codificación de Video Multivista (MVC).

[0043] Recientemente, el diseño de un nuevo estándar de video de codificación, es decir, la codificación de video de alta eficiencia (HEVC), ha sido finalizado por el Equipo de Colaboración Conjunta en materia de Codificación de Video (JCT-VC) del Grupo de Expertos en materia de Codificación de Video (VCEG) ITU-T y el Grupo de Expertos de Imágenes en Movimiento (MPEG) ISO/IEC. Una copia del estándar de HEVC finalizado (es decir, el ITU-T H.265, Series 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-I/en, (en adelante el "Estándar de HEVC".

[0044] Una extensión del alcance de HEVC, a saber la Codificación de Contenido de Pantalla (SCC) de HEVC, también está siendo desarrollado por la JCT-VC. Un proyecto reciente de HEVC SCC (Joshi et al, "High Efficiency Video Coding (HEVC) Screen Content Coding: Proyecto 4." Equipo de Colaboración Conjunta en materia de Codificación de Video (JCT-VC) de la ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, Reunión 21: Varsovia, PL 16 de junio a 19 de junio de 2015, está disponible en http: //fenix.it- sudparis.eu/jct/doc_end_user/documents/21_Warsaw/wg11/JCTVC - U1005-v2.zip, (en lo sucesivo "Proyecto 4 de HEVC SCC").

[0045] Con respecto al marco de HEVC, como un ejemplo, las técnicas de codificación basada en paletas pueden estar configurados para ser utilizadas como un modo de unidad de codificación (CU). En otros ejemplos, las técnicas de codificación basada en paletas pueden estar configuradas para ser utilizadas como un modo de unidad de predicción (PU) en el marco de HEVC. En consecuencia, todos los siguientes procesos revelados que se describen en el contexto de un modo de CU se pueden, en forma adicional o alternativa, aplicar a PU. Sin embargo, estos ejemplos basados en HEVC no deben considerarse como una restricción o limitación de las técnicas de codificación basada en paletas descriptos en la presente memoria, ya que tales técnicas se pueden aplicar para trabajar en forma independiente o como parte de otros sistemas estándares existentes o aún por desarrollar. En estos casos, la unidad para la codificación por paletas pueden ser bloques cuadrados, bloques rectangulares, o incluso regiones de forma no rectangular.

[0046] La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de video del sistema 10 que puede utilizar las técnicas de codificación de esta revelación. De

acuerdo con lo utilizado en la presente memoria, el término "codificador de video" se refiere en forma genérica tanto a codificadores de video como a decodificadores de video. En esta revelación, los términos "codificación de video" o "codificación" se pueden referir en forma genérica a la codificación de video o a la decodificación de video. El codificador de video 20 y el decodificador de video 30 del sistema de codificación de video 10 representan ejemplos de dispositivos que se pueden configurar para llevar a cabo las técnicas de codificación de video basadas en paletas de acuerdo con diversos ejemplos que se describen en esta revelación. Por ejemplo, el codificador de video 20 y el decodificador de video 30 pueden estar configurados para codificar en forma selectiva varios bloques de datos de video, tales como CU o PU en la codificación de HEVC, por el uso de codificación basada en paletas o codificación no basada en paletas. Los modos de codificación no basados en paletas se pueden referir a diversos modos de codificación temporal de inter-predicción o modos de codificación espacial intra-predicción, tales como los diversos modos de codificación especificados por el estándar de HEVC.

[0047] De acuerdo con lo mostrado en la FIG. 1, el sistema de codificación 10 de video incluye un dispositivo de fuente 12 y un dispositivo de destino 14. El dispositivo de fuente 12 genera datos de video codificados. De acuerdo con ello, el dispositivo de fuente 12 se puede denominar como un dispositivo de codificación de video o un aparato de codificación de video. El dispositivo de destino 14 puede decodificar los datos de video codificados generados por el dispositivo de fuente 12. En consecuencia, el dispositivo de destino 14 se puede denominar como un dispositivo de decodificación de video o un aparato de decodificación de video. El dispositivo de fuente 12 y el dispositivo de destino 14 pueden ser ejemplos de dispositivos de codificación de video o aparatos de codificación de video.

[0048] El dispositivo de fuente 12 y el dispositivo de destino 14 pueden comprender una amplia gama de dispositivos, que incluyen computadoras de escritorio, dispositivos de computación móviles, computadoras notebook (por ej., computadoras laptop), computadoras tablet, decodificadores, aparatos telefónicos, tales como los teléfonos llamados "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de juegos de video, computadoras montadas en automóviles, o similares.

[0049] El dispositivo de destino 14 puede recibir datos de video codificados desde el dispositivo de fuente 12 a través de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de mover los datos de video codificados desde el

dispositivo de fuente 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 fuente 12 transmitir datos de video codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de fuente 12 puede modular los datos de video codificados de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de video modulados al dispositivo de destino 14. Los uno o más medios de comunicación pueden incluir medios de comunicación inalámbrica y/o por cable, tal como un espectro de frecuencia de radio (RF) o una o más líneas de transmisión físicas. Los uno o más medios de comunicación pueden formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia, o una red global (por ej., la Internet). Los uno o más medios de comunicación pueden incluir enrutadores, conmutadores, estaciones base, u otros equipos que facilitan la comunicación del dispositivo de fuente 12 al dispositivo de destino 14.

[0050] En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacena datos de video codificados generados por el dispositivo de fuente 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento a través de acceso al disco o acceso a la tarjeta. El medio de almacenamiento puede incluir una variedad de medios de almacenamiento de datos de acceso local, tales como los discos Blu-ray, DVD, CD-ROM, memoria flash u otros medios de almacenamiento digitales adecuados para el almacenamiento de datos de video codificados.

[0051] En un ejemplo adicional, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacena codificados los datos de video generados por el dispositivo de fuente 12. En este ejemplo, el dispositivo de destino 14 puede acceder a los datos de video codificados almacenados en el servidor de archivos u otro dispositivo de almacenamiento intermedio a través de streaming o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de video codificados y transmitir los datos de video codificados al dispositivo de destino 14. Los servidores de archivos representativos incluyen los servidores web (por ej., para un sitio web), protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados a redes (NAS) y unidades de disco locales.

[0052] El dispositivo de destino 14 puede acceder a los datos de video codificados a través de una conexión de datos estándar, tal como una conexión a Internet. Los tipos de conexiones de datos representativas pueden incluir canales inalámbricos (por ej.,

conexiones Wi-Fi), conexiones inalámbricas (por ej., DSL, cable módem, etc.), o combinaciones de ambos que son adecuadas para acceder a los datos de video codificados y almacenados en un servidor de archivos. La transmisión de datos de video codificados desde el servidor de archivos puede ser una transmisión streaming, una transmisión de descarga, o una combinación de ambos.

[0053] Las técnicas de esta revelación no se limitan a aplicaciones inalámbricas o ajustes. Las técnicas se pueden aplicar a la codificación de video en apoyo de una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video por streaming, por ejemplo, a través de Internet, la codificación de datos de video para su almacenamiento en un medio de almacenamiento de datos, la decodificación de los datos de video almacenados en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema de codificación de video 10 se puede configurar para soportar la transmisión de video de dos vías de un solo sentido o dar soporte a aplicaciones tales como la transmisión de video, la reproducción de video, la transmisión de video y/o la telefonía de video.

[0054] La FIG. 1 es simplemente un ejemplo y las técnicas de esta revelación pueden estar sujetas a las configuraciones de codificación de video (por ej., codificación de video o decodificación de video) que no necesariamente incluyen alguna comunicación de datos entre los dispositivos de codificación y decodificación. En otros ejemplos, los datos se recuperan de una memoria local, por medio de streaming a través de una red, o similares. Un dispositivo de codificación de video puede codificar y almacenar datos en la memoria, y/o un dispositivo de decodificación de video puede recuperar y decodificar los datos de la memoria. En muchos ejemplos, la codificación y la decodificación se llevan a cabo por medio de dispositivos que no se comunican entre sí, sino que simplemente codifican los datos a la memoria y/o recuperan y decodifican los datos de la memoria. El dispositivo de fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, que incluyen computadoras de escritorio, computadoras notebook (por ej., computadoras laptop), computadoras tablet, decodificadores, electrodomésticos, aparatos de telefonía, tales como los teléfonos llamados "inteligentes", las almohadillas llamadas "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de juegos de video, dispositivos de transmisión de video, o similares. En algunos casos, el dispositivo de fuente 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0055] El dispositivo de destino 14 puede recibir los datos de video codificados a ser decodificados a través de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de video codificados desde el dispositivo de fuente 12 hasta el dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir al dispositivo de fuente 12 transmitir los datos de video codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de video codificados pueden ser modulados de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, tales como una frecuencia de radio (RF) del espectro o una o más líneas de transmisión físicas. 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 enrutadores, conmutadores, estaciones base, o cualquier otro equipo que pueda ser útil para facilitar la comunicación del dispositivo de fuente 12 al dispositivo de destino 14.

[0056] Como alternativa, los datos codificados pueden ser emitidos desde la interfaz de salida 22 a un dispositivo de almacenamiento 19. Del mismo modo, los datos codificados se pueden acceder desde el dispositivo de almacenamiento 19 por medio de la interfaz de entrada. El dispositivo de almacenamiento 19 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o a los que se accede localmente tal como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado medios para el almacenamiento de datos de video codificados. En un ejemplo adicional, el dispositivo de almacenamiento 19 puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede contener el video codificado generado por el dispositivo de fuente 12. El dispositivo de destino 14 puede acceder a los datos de video almacenados desde el dispositivo de almacenamiento 19 a través de streaming o descargas. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de video y transmitir esos datos de video codificados al dispositivo de destino 14. Los servidores de archivos representativos incluyen un servidor web (por ej., para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a redes (NAS), o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de video codificados a través de cualquier conexión de datos estándar, que incluyen una conexión a Internet.

Esto puede incluir un canal inalámbrico (por ej., una conexión Wi-Fi), una conexión por cable (por ej., DSL, módem de cable, etc.), o una combinación de ambos que es adecuada para acceder a los datos de video codificados y almacenados en un servidor de archivos. La transmisión de datos de video codificados desde el dispositivo de almacenamiento 19 puede ser una transmisión streaming, una transmisión de descarga, o una combinación de ambos.

[0057] Las técnicas de esta revelación no se limitan necesariamente a aplicaciones inalámbricas o ajustes. Las técnicas se pueden aplicar a la codificación de video en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, cable, transmisiones de televisión por satélite, transmisiones de video por streaming, por ej., a través de Internet, la codificación de la tecnología digital de video para su almacenamiento en un medio de almacenamiento de datos, la decodificación de video digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 se puede configurar para soportar la transmisión de video de dos vías de un solo sentido o dar soporte a aplicaciones tales como la transmisión de video, la reproducción de video, la transmisión de video y/o la telefonía de video.

[0058] En el ejemplo de la FIG. 1, el dispositivo de fuente 12 incluye una fuente de video 18, el codificador de video 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo de fuente 12, la fuente de video 18 puede incluir una fuente tal como un dispositivo de captura de video, por ejemplo, una cámara de video, un archivo de video que contiene video previamente capturado, una interfaz de entrada de video para recibir el video de un proveedor de contenido de video, y/o un sistema de gráficos de computadora para generar datos de gráficos de computadora como el video de fuente, o una combinación de tales fuentes. Como un ejemplo, si la fuente de video 18 es una cámara de video, el dispositivo de fuente 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o video teléfonos. Sin embargo, las técnicas descriptas en esta revelación se pueden aplicar a la codificación de video en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas.

[0059] El video capturado, pre-capturado, o generado por computadora puede ser codificado por el codificador de video 20. Los datos de video codificados pueden ser transmitido directamente al dispositivo de destino 14 a través de la interfaz de salida 22 del dispositivo de fuente 12. Los datos de video codificados también (o en forma alternativa) se pueden almacenar en el dispositivo de almacenamiento 19 para su

posterior acceso por el dispositivo de destino 14 u otros dispositivos, para la decodificación y/o la reproducción.

[0060] El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de video 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 de dispositivo de destino 14 recibe los datos de video codificados por el enlace 16. Los datos de video codificados comunicados a través del enlace 16, o suministrado en el dispositivo de almacenamiento 19, puede incluir una variedad de elementos de sintaxis generados por el codificador de video 20 para su uso por un decodificador de video, tal como el decodificador de video 30, en la decodificación de los datos de video. Tales elementos de sintaxis pueden estar incluidos con los datos de video codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

[0061] El dispositivo de visualización 32 puede estar integrada con, o en su exterior, 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. Por lo general, el dispositivo de visualización 32 muestra los datos de video decodificados a 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 diodo orgánico emisor de luz (OLED), u otro tipo de dispositivo de visualización.

[0062] El codificador de video 20 y el decodificador de video 30 pueden funcionar de acuerdo con un estándar de compresión de video, tales como el estándar de HEVC recientemente finalizado (y varias extensiones del mismo actualmente en fase de desarrollo). En forma alternativa, el codificador de video 20 y el decodificador de video 30 pueden funcionar de acuerdo con otros estándares propietarios o de la industria, tales como el estándar ITU-T H.264, denominado en forma alternativa como MPEG4, Parte 10, Codificación de Video Avanzada (AVC), o las extensiones de tales estándares. Las técnicas de esta revelación, sin embargo, no se limitan a cualquier estándar de codificación particular. Otros ejemplos de estándares de compresión de video incluyen VP8, y VP9.

[0063] Si bien no se muestra en la FIG. 1, en algunos aspectos, el codificador de video 20 y el decodificador de video 30 pueden cada uno estar integrados con un codificador y un decodificador de audio, y pueden incluir unidades apropiadas MUX-DEMUX, u

otro hardware y software, para manejar la codificación de audio y video en una corriente de datos común o corrientes de datos separados. En su caso, en algunos ejemplos, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor H.223 de la ITU, u otros protocolos tales como el protocolo de datagramas de usuario (UDP).

[0064] Cada uno del codificador de video 20 y el decodificador de video 30 se puede implementar como cualquiera de una variedad de circuitos codificadores adecuados, tal como uno o más circuitos integrados que incluyen microprocesadores, procesadores de señal digital (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables de campo (FPGA), lógica discreta, software, hardware, firmware, o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar las instrucciones para el software en un medio legible por computadora no transitorio adecuado, y ejecutar las instrucciones de hardware tales como circuitos integrados por el uso de uno o más procesadores para llevar a cabo las técnicas de esta revelación. Cada uno del codificador de video 20 y el decodificador de video 30 se pueden incluir en uno o más codificadores o decodificadores, cualquiera de los cuales pueden ser integrados como parte de un codificador/decodificador combinado (CÓDEC) en un dispositivo respectivo.

[0065] De acuerdo con lo introducido con anterioridad, el JCT-VC ha finalizado recientemente el desarrollo del estándar de HEVC. Los esfuerzos de normalización HEVC se basaron en un modelo en evolución de un dispositivo de codificación de video que se denomina como el Modelo de Prueba de HEVC (HM). El HM presume varias capacidades adicionales de dispositivos de codificación de video relativos a los dispositivos existentes de acuerdo con, por ejemplo, el ITU-T H.264/AVC. Por ejemplo, mientras que el H.264 proporciona nueve modos de codificación de intra-predicción, el HM puede proporcionar hasta treinta y cinco modos de codificación de intra-predicción.

[0066] En HEVC y otras especificaciones de codificación de video, una secuencia de video incluye en forma típica una serie de imágenes. Las imágenes también se pueden denominar como "tramas". Una imagen puede incluir tres matrices de muestra, denotadas como S_L, S_{Cb}, y S_{Cr}. S_L es una matriz de dos dimensiones (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 la presente memoria como muestras de "croma". En otros casos, una imagen puede ser monocromática e incluir

sólo una matriz de muestras de luma.

[0067] Para generar una representación codificada de una imagen, el codificador de video 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 las muestras de luma, dos bloques de árbol de codificación correspondientes de muestras de croma, y estructuras de sintaxis que se utilizan 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 codificación de árbol y estructuras de sintaxis que se utilizan para codificar las muestras del bloque de árbol de codificación. Un bloque de árbol de codificación puede ser un bloque de NxN muestras. Una CTU también se puede denominar como un "bloque de árbol" o una LCU. La CTU de HEVC puede ser ampliamente análogo a los macrobloques de otros estándares, tales como H.264/AVC. Sin embargo, una CTU no se limita necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). Un corte puede incluir un número entero de CTU ordenado en forma consecutiva en un orden de exploración de trama.

[0068] Para generar una CTU codificada, el codificador de video 20 puede llevar a cabo un particionamiento en árbol cuádruple de forma recursiva 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 de "unidades de árbol de codificación". Un bloque de codificación puede ser un bloque de NxN de muestras. Una CU puede comprender un bloque de codificación de las muestras de luma y dos bloques de muestra de croma correspondientes 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 las estructuras de sintaxis que se utilizan para codificar las muestras de los bloques de codificación. En las imágenes monocromáticas o imágenes que tienen tres planos de color separados, una CU puede comprender un único bloque de codificación y estructuras de sintaxis que se utilizan para codificar las muestras del bloque de codificación.

[0069] El codificador de video 20 puede particionar un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrada o no cuadrada) 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 las muestras de luma, dos bloques de predicción correspondientes de muestras de croma, y estructuras de sintaxis que se utilizan 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 de sintaxis que se utilizan para predecir el bloque de predicción. El codificador de video 20 puede generar bloques de predicción luma, Cb, y Cr bloques de predicción luma, Cb, Cr de cada PU de la CU.

[0070] El codificador de video 20 puede utilizar la intra-predicción o inter predicción para generar los bloques de predicción para una PU. Si el codificador de video 20 utiliza la intra-predicción para generar los bloques de predicción de una PU, el codificador de video 20 puede generar los bloques de predicción de la PU con base en muestras decodificadas de la imagen asociada con la PU. Si el codificador de video 20 utiliza inter-predicción para generar los bloques de predicción de una PU, el codificador de video 20 puede generar los bloques de predicción de la PU con base en muestras decodificadas de una o más imágenes distintas de la imagen asociada con la PU.

[0071] Después de que el codificador de video 20 genera bloques de predicción luma, Cb, y Cr para una o más unidades PU de una CU, el codificador de video 20 puede generar un bloque residual luma para la CU. Cada muestra en el bloque residual luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques de predicción luma de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de video 20 puede generar un bloque residual Cb para la CU. Cada muestra en el bloque residual Cb de la CU puede indicar una diferencia entre una muestra Cb en uno de los bloques Cb predictivos de la CU y una muestra correspondiente en el bloque de codificación original de Cb de la CU. El codificador de video 20 también puede generar un bloque residual Cr para la CU. Cada muestra en el bloque residual Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques de predicción Cr de la CU y una muestra correspondiente en el bloque de codificación Cr original de la CU.

[0072] Además, el codificador de video 20 puede utilizar el particionamiento en árbol cuádruple para descomponer los bloques residuales luma, Cb y Cr de una CU en uno o más bloques de transformada luma, Cb, y Cr. Un bloque de transformada es una forma rectangular (por ej., cuadrada o no cuadrada) de muestras sobre el que se aplica la misma transformación. Una unidad de transformada (TU) de una CU puede comprender un bloque de transformada de muestras de luma, dos bloques de transformada correspondientes de muestras de croma, y las estructuras de sintaxis que se utilizan para transformar las muestras de bloque de transformada. Por lo tanto,

cada TU de una CU puede estar asociada con un bloque de transformada de luma, un bloque de transformada Cb, y un bloque de transformada Cr. El bloque de transformada luma asociado con la TU puede ser un sub-bloque del bloque residual luma de la CU. El bloque de transformada Cb puede ser un sub-bloque del bloque residual Cb de la CU. El bloque de transformada Cr puede ser un sub-bloque del bloque residual 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 las estructuras de sintaxis que se utilizan para transformar las muestras del bloque de transformada.

[0073] El codificador de video 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 video 20 puede aplicar una o más transformadas a un bloque de transformada Cb de una TU para generar un bloque de coeficientes Cb para la TU. El codificador de video 20 puede aplicar una o más transformadas a un bloque de transformada Cr de una TU para generar un bloque de coeficientes Cr para la TU.

[0074] Después de generar un bloque de coeficientes (por ej., un bloque de coeficientes luma, un bloque de coeficientes Cb o un bloque de coeficientes Cr), el codificador de video 20 puede cuantificar el bloque de coeficientes. La cuantificación se refiere por lo general a un proceso en el que los coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos utilizados para representar los coeficientes de transformada, por medio del suministro de una compresión adicional. Después de que el codificador de video 20 cuantifica un bloque de coeficientes, el codificador de video 20 puede codificar por entropía elementos de sintaxis que indican los coeficientes de transformada cuantificados. Por ejemplo, el codificador de video 20 puede llevar a cabo una Codificación Aritmética Binaria Adaptativa al Contexto (CABAC) en los elementos de sintaxis que indican los coeficientes de transformada cuantificados.

[0075] El codificador de video de salida 20 puede dar salida a 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 de sintaxis que contiene una indicación del tipo de datos en la unidad NAL y bytes que contienen los datos en la forma de un RBSP intercalado de acuerdo con lo necesario con bits de prevención de

la emulación. Cada una de las unidades NAL incluye una cabecera de unidad NAL y encapsula un RBSP. La cabecera de unidad NAL puede incluir un elemento de sintaxis que indica un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por la cabecera de unidad NAL de una unidad NAL indica el tipo de la unidad NAL. Un RBSP puede ser una estructura de sintaxis que contiene un número entero de bytes que está encapsulado dentro de una unidad NAL. En algunos casos, un RBSP incluye cero bits.

[0076] Los diferentes tipos de unidades NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad NAL puede encapsular un RBSP de PPS, un segundo tipo de unidad NAL puede encapsular un RBSP para un corte codificada, un tercer tipo de unidad NAL puede encapsular un RBSP para mensajes SEI, y así sucesivamente. Las unidades NAL que encapsulan RBSP para la codificación de datos de video (en oposición a los RBSP para conjuntos de parámetros y mensajes SEI) se pueden denominar como unidades NAL de VCL.

[0077] El decodificador de video 30 puede recibir una corriente de bits generada por el codificador de video 20. Además, el decodificador de video 30 puede analizar la corriente de bits para obtener elementos de sintaxis de la corriente de bits. El decodificador de video 30 puede reconstruir las imágenes de los datos de video con base por lo menos en parte, en los elementos de sintaxis obtenidos a partir de la corriente de bits. El proceso para reconstruir los datos de video por lo general puede ser recíproco al proceso llevado a cabo por el codificador de video 20. Además, el decodificador de video 30 puede cuantificar en forma inversa los bloques de coeficientes asociados con las TU de una CU actual. El decodificador de video 30 puede llevar a cabo transformadas inversas de los bloques de coeficientes para reconstruir bloques de transformada asociados con las TU de la CU actual. El decodificador de video 30 puede reconstruir los bloques de codificación de la CU actual por medio de la adición de las muestras de los bloques de predicción para las PU de la CU actual para transformar las muestras de bloques de las TU de la CU actual correspondiente. Por medio de la reconstrucción de los bloques de codificación para cada CU de una imagen, el decodificador de video 30 puede reconstruir la imagen.

[0078] En algunos ejemplos, el codificador de video 20 y el decodificador de video 30 pueden estar configurados para llevar a cabo la codificación basada en paletas. Por ejemplo, en la codificación basada en paletas, en lugar de llevar a cabo las técnicas de codificación de intra-predicción o inter-predicción descriptas con anterioridad, el

codificador de video 20 y el decodificador de video 30 pueden codificar una denominada paleta como una tabla de valores de color para la representación de los datos de video del área particular (por ej., un bloque dado). Cada píxel puede estar asociado con una entrada en la paleta que representa el color del píxel, por ejemplo, con un valor de luma (Y) y valores de croma (Cb y Cr). Por ejemplo, el codificador de video 20 y el decodificador de video 30 pueden codificar un índice que relaciona el valor de píxel con el valor apropiado en la paleta.

[0079] En el ejemplo anterior, el codificador de video 20 puede codificar un bloque de datos de video por medio de la determinación de una paleta para el bloque, la localización de una entrada en la paleta para representar el valor de cada píxel, y la codificación por paletas con los valores de índice para los píxeles que relacionan el de píxel con la paleta. El decodificador de video 30 puede obtener, a partir de una corriente de bits codificada, una paleta para un bloque, así como también los valores de índice para los píxeles del bloque. El decodificador de video 30 puede relacionar los valores de índice de los píxeles de las entradas de paleta para reconstruir los valores de píxel del bloque.

[0080] Los aspectos de esta revelación están dirigidos a la derivación de paletas, que puede ocurrir en el codificador y en el decodificador. Como un ejemplo, el codificador de video 20 puede derivar una paleta para un bloque actual por medio de la derivación de un histograma de los píxeles en el bloque actual. En algunos ejemplos, el histograma se puede expresar como $H = \{(v_i, f_i), i = \{0,1,2,\cdots,M\}\}$ donde $M + 1$ es el número de diferentes valores de píxel en el bloque actual, v_i es valor de píxel, y f_i es el número de aparición de v_i (es decir, el número de píxeles en el bloque actual tiene un valor de píxel v_i). En tales ejemplos, el histograma representa por lo general un número de veces que un valor de píxel se produce en el bloque actual.

[0081] El codificador de video 20 puede inicializar una o más variables cuando se deriva el histograma. Como un ejemplo, el codificador de video 20 puede inicializar un índice de paleta *idx* a 0, (es decir, establecer *idx* = 0). Como otro ejemplo, el codificador de video 20 puede inicializar la paleta P para estar vacía (es decir, $P = \emptyset$, establecer j = 0.).

[0082] El codificador de video 20 puede ordenar el histograma, por ej., en orden descendente, de manera tal que los píxeles que tienen más apariciones se colocan cerca de la parte delantera de una lista de valores. Por ejemplo, el codificador de video 20 puede ordenar H de acuerdo con el orden descendente de *f* y la lista ordenada se

puede expresar como $H_o = \{(u_i, f_i), i = \{0,1,2,\cdots,M\}, f_i \geq f_{i+1}\}$. En este ejemplo, la lista ordenada incluye los valores de píxel que se producen con mayor frecuencia en la parte delantera (superior) de la lista y los valores de píxel que se producen con menor frecuencia en la parte trasera (parte inferior) de la lista.

[0083] El codificador de video 20 puede copiar una o más entradas a partir del histograma en la paleta. Como un ejemplo, el codificador de video 20 puede insertar la entrada en el histograma con la mayor frecuencia en la paleta. Por ejemplo, el codificador de video 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 video 20 puede evaluar la entrada en el histograma con la siguiente mayor frecuencia para la inserción en la paleta. Por ejemplo, el codificador de video 20 puede establecer idx = idx + 1, j = j + 1.

[0084] El codificador de video 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) en la paleta (es decir, Distancia (u_{j+1}, x) < Thresh). Por ejemplo, el codificador de video 20 puede determinar si la entrada está dentro de la vecindad de cualquier píxel de la paleta por medio de la determinación de si un valor de la entrada se encuentra dentro de una distancia umbral de un valor de cualquier píxel de la paleta. En algunos ejemplos, el codificador de video 20 puede seleccionar en forma flexible la función de distancia. Como un ejemplo, el codificador de video 20 puede seleccionar la función de distancia como una suma de las diferencias absolutas (SAD) o una suma de errores cuadráticos de predicción (SSE) de los tres componentes de color (por ej., cada uno de luminancia, crominancia de matiz azul, y crominancia de matiz rojo), o un componente de color (por ej., uno de luminancia, crominancia de matiz azul, o crominancia de matiz rojo). En algunos ejemplos, el codificador de video 20 puede seleccionar en forma flexible el valor de umbral Thresh. Como un ejemplo, el codificador de video 20 puede seleccionar el valor de umbral a ser dependiente del parámetro de cuantificación (QP) del bloque actual. Como otro ejemplo, el codificador de video 20 puede seleccionar el valor de umbral para ser dependiente del valor de *idx* o el valor de *j*.

[0085] Si el codificador de video 20 determina que la entrada con la siguiente mayor frecuencia (es decir, u_{j+1}) se encuentra dentro de la vecindad de cualquier píxel de la paleta, el codificador de video 20 no puede insertar la entrada en el histograma. Si el codificador de video 20 determina que la entrada con la siguiente mayor frecuencia (es decir, u_{j+1}) no está dentro de la vecindad de cualquier píxel de la paleta, el codificador

de video 20 puede insertar la entrada en el histograma.

[0086] El codificador de video 20 puede continuar insertando entradas en la paleta hasta que se satisfagan una o más condiciones. Algunas condiciones representativas son cuando idx = M, cuando j = M, o cuando el tamaño de la paleta es mayor que un valor predefinido.

[0087] La codificación basada en paletas puede tener una cierta cantidad de sobrecarga de señalización. Por ejemplo, un número de bits puede ser necesario para indicar características de una paleta, tales como un tamaño de la paleta, así como también la paleta en sí mismo. Además, puede ser necesario un número de bits para indicar los valores de índice para los píxeles del bloque. Las técnicas de la presente revelación pueden, en algunos ejemplos, reducir el número de bits necesarios para indicar dicha información. Por ejemplo, las técnicas descriptas en esta revelación pueden incluir técnicas para varias combinaciones de uno o más modos de codificación basado en paletas de señalización, transmisión por paletas, predicción por paletas, derivación por paletas, y la transmisión de mapas de codificación basados en paletas y otros elementos de sintaxis.

[0088] En algunos ejemplos, el codificador de video 20 y/o el decodificador de video 30 pueden predecir una paleta con otra paleta. Por ejemplo, el codificador de video 20 y/o el decodificador de video 30 pueden determinar una primera paleta que tiene una primera información que indica los primeros valores de píxel. El codificador de video 20 y/o el decodificador de video 30 pueden entonces determinar, con base en las primeras entradas de la primera paleta, una o más segundas entradas que indican segundos valores de píxel de una segunda paleta. El codificador de video 20 y/o el decodificador de video 30 también puede codificar píxeles de un bloque de datos de video por el uso de la segunda paleta.

[0089] Cuando se determinan las entradas de la segunda paleta con base en las entradas de la primera paleta, el codificador de video 20 puede codificar una variedad de elementos de sintaxis, que pueden ser utilizados por el decodificador de video para reconstruir la segunda paleta. Por ejemplo, el codificador de video 20 puede codificar uno o más elementos de sintaxis en una corriente de bits para indicar que una paleta entera (o paletas, en el caso de cada componente de color, por ej., Y, Cb, Cr, o Y, U, V, o R, G, B, de los datos de video que tienen una paleta por separado) se copia de uno o más bloques vecinos del bloque que está siendo codificado. La paleta de la que se predicen las entradas de paleta actual del bloque actual (por ej., copiar) se puede denominar como una paleta de predicción. La paleta de predicción puede contener

entradas de paleta a partir de uno o más bloques vecinos, incluidos los bloques espacialmente vecinos y/o bloques vecinos en un orden de exploración en particular de los bloques. Por ejemplo, los bloques vecinos pueden estar ubicados en forma espacial a la izquierda (bloque vecino izquierdo) de o superior (bloque vecino superior) del bloque que está siendo codificado. En otro ejemplo, el codificador de video 20 puede determinar las entradas de paleta de predicción por el uso de los valores de las muestras más frecuentes en un vecino de causalidad del bloque actual. En otro ejemplo, los bloques vecinos pueden ser vecinos del bloque actual que está siendo codificado de acuerdo con un orden de exploración en particular utilizado 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 video 20 puede codificar uno o más elementos de sintaxis para indicar la ubicación de los bloques de vecinos de los cuales se copian las paletas.

[0090] En algunos ejemplos, la predicción por paletas se puede llevar a cabo entrada a entrada. Por ejemplo, el codificador de video 20 puede codificar uno o más elementos de sintaxis para indicar, para cada entrada de una paleta de predicción, si la entrada de paleta está incluida en la paleta para el bloque actual. Si el codificador de video 20 no predice una entrada de paleta para el bloque actual, el codificador de video 20 puede codificar uno o más elementos de sintaxis para especificar las entradas no predichas, así como también el número de tales entradas.

[0091] Los elementos de sintaxis descriptos con anterioridad se pueden denominar como un vector de predicción por paletas. Por ejemplo, de acuerdo con lo señalado con anterioridad, el codificador de video 20 y el decodificador de video 30 pueden predecir una paleta para un bloque actual con base en una o más paletas de bloques vecinos (denominada en forma colectiva como una paleta de referencia). Cuando se genera la paleta de referencia, se puede utilizar un criterio de primero en entrar, primero en salir (FIFO) por medio de la adición de la última paleta en la parte delantera de la cola. Si la cola excede un umbral predefinido, los elementos más antiguos se pueden hacer estallar hacia fuera. Después de empujar nuevos elementos en la parte delantera de la cola, se puede aplicar un proceso de poda para eliminar los elementos duplicados, a contar desde el comienzo de la cola. En concreto, en algunos ejemplos, el codificador de video 20 puede codificar (y el decodificador de video 30 puede decodificar) un vector de 0-1 para indicar si los valores de píxel de la paleta de referencia se vuelven a utilizar para la paleta actual. Como un ejemplo, de acuerdo con lo mostrado en el ejemplo de la Tabla 6, una paleta de referencia puede incluir seis

elementos (por ej., seis valores de índice y valores de píxel respectivos).

Índice	Valor de píxel
0	v_0
1	v_1
2	v_2
3	v_3
4	v_4
5	v_5

Tabla 6

En un ejemplo con fines de ilustración, el codificador de video 20 puede señalizar un vector (1, 0, 1, 1, 1, 1) que indica que v_0, v_2, v_3, v_4 y v_5 se vuelve a utilizar en la paleta actual, mientras que v_1 no se reutiliza. Además de la reutilización de v_0, v_2, v_3, v_4, y v_5, el codificador de video 20 puede añadir dos nuevos elementos a la paleta actual con los índices 5 y 6. La paleta actual para este ejemplo se muestra en la Tabla 7, a continuación.

Marcador Pred.	Índice	Valor de píxel
1	0	v0
0		
1	1	v_2
1	2	v_3
1	3	v_4
1	4	v_5
	5	u_0
	6	u_1

Tabla 7

[0092] Para codificar el vector de predicción por paletas de 0-1, para cada elemento en el vector, el codificador de video 20 puede codificar un bit para representar su valor. Además, el número de elementos de paleta que no se puede predecir (por ej., el número de nuevas entradas de paleta (u0 y u1 en el ejemplo de la Tabla 7 más arriba)) se puede binarizar y señalizar.

[0093] Otros aspectos de esta revelación se refieren a la construcción y/o transmisión de un mapa que permite que el codificador de video 20 y/o el decodificador de video 30 determine los valores de píxel. Por ejemplo, otros aspectos de esta revelación se refieren a la construcción y/o transmisión de un mapa de índices que relacionan un píxel particular a una entrada de una paleta.

[0094] En algunos ejemplos, el codificador de video 20 puede indicar si los píxeles de un bloque tienen un valor correspondiente en una paleta. En un ejemplo con fines de ilustración, si se supone que una entrada (i, j) de un mapa corresponde a una posición de píxel (i, j) en un bloque de datos de video. En este ejemplo, el codificador de video 20 puede codificar un marcador para cada posición de píxel de un bloque. El codificador de video 20 puede establecer el indicador igual a uno para la entrada (i, j) para indicar que el valor del píxel en la ubicación (i, j) es uno de los valores de la paleta. Cuando un color se incluye en la paleta (es decir, el marcador es igual a uno), el codificador de video 20 también puede codificar los datos que indican un índice de paleta para la entrada (i, j) que identifica el color de la paleta. Cuando el color del píxel no está incluido en la paleta (es decir, el marcador es igual a cero) el codificador de video 20 también puede codificar datos que indican un valor de muestra para el píxel, que se puede denominar como un píxel de escape. El decodificador de video 30 puede obtener los datos descriptos con anterioridad a partir de una corriente de bits codificada y utilizar los datos para determinar un índice de paleta y/o el valor del píxel para un lugar determinado en un bloque.

[0095] En algunos casos, puede haber una correlación entre el índice de paleta a la que se asigna un píxel en una posición dada y la probabilidad de un píxel vecino que se asigna al mismo índice de paleta. Es decir, cuando un píxel se asigna a un índice de paleta particular, la probabilidad puede ser relativamente alta de que uno o más píxeles vecinos (en términos de localización espacial) se asigne al mismo índice de paleta.

[0096] En algunos ejemplos, el codificador de video 20 y/o el decodificador de video 30 pueden determinar y codificar uno o más índices de un bloque de datos de video en relación con uno o más índices del mismo bloque de datos de video. Por ejemplo, el codificador de video 20 y/o el decodificador de video 30 pueden estar configurados para determinar un primer valor de índice asociado con un primer píxel en un bloque de datos de video, en el que el primer valor de índice relaciona un valor del primer píxel con una entrada de una paleta. El codificador de video 20 y/o el decodificador de video 30 también puede estar configurado para determinar, con base en el primer valor de índice, uno o más segundos valores de índice asociados con uno o más segundos píxeles en el bloque de datos de video, y para codificar el primer y los uno o más segundos píxeles del bloque de datos de video. Por lo tanto, en este ejemplo, los índices de un mapa se pueden codificar con respecto a uno o más índices del mapa.

[0097] De acuerdo con lo discutido con anterioridad, el codificador de video 20 y/o el

decodificador de video 30 pueden utilizar varias técnicas diferentes para codificar los valores de índice de un mapa con relación a otros índices del mapa. Por ejemplo, el codificador de video 20 y/o el decodificador de video 30 pueden utilizar el modo de índice, el modo de copia desde arriba, y el modo de transición para codificar los valores de índice de un mapa en relación con otros índices del mapa.

[0098] En el "modo de índice" de codificación basada en paletas, el codificador de video 20 y/o el decodificador de video 30 pueden señalizar 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 video 20 y/o el decodificador de video 30 pueden indicar el valor de valor de la muestra o muestras cuantificadas para cada componente. Por ejemplo, si el tamaño de la paleta es 4, para muestras no de escape, los índices de paleta están en el intervalo [0, 3]. En este caso, un valor de índice 4 puede significar una muestra de escape. Si el índice indica una muestra no de escape, el codificador de video 20 y/o el decodificador de video 30 pueden señalizar una longitud de ejecución, que puede especificar el número de muestras subsiguientes en el orden de exploración que comparten el mismo índice, por un valor de ***n-1*** no negativo indica la longitud de ejecución, lo que significa que los siguientes ***n*** píxeles incluido el actual tienen el mismo índice de píxel que el primer índice señalado.

[0099] En la modo de codificación basado en paletas de "copia desde arriba", el codificador de video 20 y/o el decodificador de video 30 pueden señalizar un valor de longitud de ejecución no negativo ***m-1*** para indicar que para los siguientes ***m*** píxeles que incluyen el píxel actual, los índices de paleta son los mismos que sus vecinos desde arriba, respectivamente. Se debe tener en cuenta que el modo de “copia desde arriba" es diferente del modo de "índice", en el sentido de que los índices de paleta podrían ser diferentes dentro del modo de ejecución de "copia desde arriba".

[00100] De acuerdo con lo discutido con anterioridad, en algunos ejemplos, puede ser deseable agrupar los compartimientos de derivación juntos (es decir, para aumentar el rendimiento de CABAC). De acuerdo con una o más técnicas de esta revelación, el codificador de video 20 puede codificar y el decodificador de video 30 puede decodificar, los elementos de sintaxis que se utilizan para definir una paleta actual de manera tal que los elementos de sintaxis que se codifican por el uso del modo de derivación se agrupen. Por ejemplo, a diferencia de la codificación de uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para un bloque actual de datos de video antes de un elemento de sintaxis que indica si un proceso de transposición se aplica a

índices de paleta de una paleta para el bloque actual de datos de video, el codificador de video 20 y/o el decodificador de video 30 pueden codificar los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video después de que el elemento de sintaxis que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de video. De este modo, el codificador de video 20 y/o el decodificador de video 30 pueden codificar un grupo mayor de elementos de sintaxis por el uso del modo de derivación, lo que puede aumentar el rendimiento de CABAC.

[00101] En algunos ejemplos, los uno o más elementos de sintaxis relacionados con QP delta para el bloque actual de datos de video pueden incluir una serie de elementos de sintaxis que especifica el valor absoluto de la diferencia entre un QP luma para el bloque actual de datos de video y un predictor del QP luma para el bloque actual (por ej., **cu_qp_delta_palette_abs**), y un elemento de sintaxis que especifica un signo de la diferencia entre el QP luma para el bloque actual de datos de video y el predictor del QP luma para el bloque actual (por ej., **cu_qp_delta_palette_sign_flag**). En algunos ejemplos, los uno o más elementos de sintaxis relacionados con compensaciones de QP croma para el bloque actual de datos de video pueden incluir un elemento de sintaxis que indica si las entradas en una o más listas de compensaciones se añaden al QP luma para el bloque actual para determinar los QP croma para el bloque actual (por ej., **cu_chroma_qp_palette_offset_flag**) y un elemento de sintaxis que especifica un índice de una entrada en cada una de las una o más listas de compensaciones que se agregan al QP luma para el bloque actual para determinar los QP croma para el bloque actual (por ej., **cu_chroma_qp_palette_offset_idx**). Como tal, el codificador de video 20 y/o el decodificador de video 30 pueden cada uno estar configurado para codificar un elemento de sintaxis **palette_transpose_flag** en una primera posición en una corriente de bits y codificar un elemento de sintaxis **cu_qp_delta_palette_abs**, un elemento de sintaxis **cu_qp_delta_palette_sign_flag**, un elemento de sintaxis **cu_chroma_qp_palette_offset_flag**, y un elemento de sintaxis **cu_chroma_qp_palette_offset_idx** en una segunda posición en la corriente de bits que es posterior a la primera posición.

[00102] La FIG. 2 es un diagrama de bloques que ilustra un codificador de video 20 representativo que puede poner en práctica las técnicas de esta revelación. La FIG. 2 se proporciona para los propósitos de explicación y no se debe considerar limitante de las técnicas de acuerdo con lo ampliamente ejemplificado y descripto en esta

revelación. Para fines de explicación, esta revelación describe el codificador de video 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta revelación pueden ser aplicables a otros estándares o métodos de codificación.

[00103] El codificador de video 20 representa un ejemplo de un dispositivo que se puede configurar para llevar a cabo las técnicas de codificación de video basadas en paletas de acuerdo con diversos ejemplos que se describen en esta revelación. Por ejemplo, el codificador de video 20 puede estar configurado para codificar selectivamente varios bloques de datos de video, tales como CU o PU en la codificación en HEVC, por el uso de ya sea codificación basada en paletas o codificación no basada en paletas. Los modos de codificación no basados en paletas se pueden referir a diversos modos de codificación inter-temporales de predicción o modos de codificación espacial intra-predicción, tales como los diversos modos de codificación especificados por el estándar de HEVC. El codificador de video 20, en un ejemplo, puede estar configurado para generar una paleta que tiene entradas que indican los valores de píxel, seleccionar valores de píxel en una paleta para representar valores de píxel de por lo menos algunas posiciones de un bloque de datos de video, y señalizar la información asociada con por lo menos algunas de las posiciones del bloque de datos de video con entradas en la paleta correspondientes, respectivamente, a los valores de píxel seleccionados. La información señalizada puede ser utilizada por el decodificador de video 30 para decodificar datos de video.

[00104] En el ejemplo de la FIG. 2, el codificador de video 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, un búfer de imagen decodificada 116, y una unidad de codificación de la entropía 118. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de inter-predicción 120 y una unidad de procesamiento de intra-predicción 126. La unidad de procesamiento de inter-predicción 120 incluye una unidad de estimación de movimiento y una unidad de compensación de movimiento (no se muestran). El codificador de video 20 también incluye una unidad de codificación basada en paletas 122 configurada para llevar a cabo varios aspectos de las técnicas de codificación basada en paletas que se describen en esta revelación. En otros ejemplos, el codificador de video 20 puede incluir más o menos, o diferentes componentes funcionales.

[00105] El codificador de video 20 puede recibir datos de video. El codificador de video 20 puede codificar cada CTU en un corte de una imagen de los datos de video. Cada una de las CTU pueden estar asociadas con los bloques de árbol de codificación de luma de igual tamaño (CTB) y los correspondientes CTB de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 100 puede llevar a cabo un particionamiento en árbol cuádruple para dividir los CTB de la CTU en bloques progresivamente más pequeños. El bloque más pequeño puede ser bloques de codificación de CU. Por ejemplo, la unidad de procesamiento de predicción 100 puede particionar un CTB asociado a una CTU en cuatro sub-bloques de igual tamaño, particionar uno o más de los sub-bloques en cuatro sub-sub-bloques de igual tamaño, y así sucesivamente.

[00106] El codificador de video 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 unidades PU de la CU. Por lo tanto, cada PU puede estar asociada con un bloque de predicción luma y bloques de predicción croma correspondientes. El codificador de video 20 y el decodificador de video 30 pueden soportar las PU que tienen diversos tamaños. De acuerdo con lo indicado con anterioridad, el tamaño de una CU se puede referir al tamaño del bloque de codificación de luma de la CU y el tamaño de una PU se puede referir al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el tamaño de una CU particular es 2Nx2N, el codificador de video 20 y el decodificador de video 30 pueden soportar tamaños de PU de 2Nx2N o NxN para la intra-predicción y tamaños simétricos de PU de 2Nx2N, 2NxN, Nx2N, NxN, o similares para la inter-predicción. El codificador de video 20 y el decodificador de video 30 también pueden apoyar la partición asimétrica para los tamaños de PU de 2NxnU, 2NxnD, nLx2N, y nRx2N para inter-predicción.

[00107] La unidad de procesamiento de inter-predicción 120 puede generar datos de predicción de la PU por medio de la realización de una inter-predicción en cada PU de una CU. Los datos de predicción de la PU puede incluir un bloque de muestra de predicción de la PU y la información de movimiento de la PU. La unidad de procesamiento de inter-predicción 120 puede llevar a cabo diferentes operaciones para una PU de una CU dependiendo de si la PU se encuentra en un corte I, un corte P, o un corte B. En un corte I, todas las PU son de intra predicción. Por lo tanto, si la PU se encuentra en un corte I, la unidad de procesamiento de inter-predicción 120 no

lleva a cabo inter-predicción en la PU. Por lo tanto, para los bloques codificados en modo I, el bloque previsto está formado por el uso de la predicción espacial de bloques vecinos codificados previamente dentro de la misma trama.

[00108] Si una PU se encuentra en un corte P, la unidad de estimación de movimiento de la unidad de procesamiento de inter-predicción 120 puede buscar las imágenes de referencia en una lista de imágenes de referencia (por ej., "RefPicList0") para una región de referencia para la PU. La región de referencia para la PU puede ser una región, dentro de una imagen de referencia, que contiene bloques de muestra que corresponden más estrechamente a los bloques de muestra de la PU. La unidad de estimación de movimiento puede generar un índice de referencia que indica una posición en RefPicList0 de la imagen de referencia que contiene la región de referencia para la PU. Además, la unidad de estimación de movimiento puede generar un MV que indica un desplazamiento espacial entre un bloque de codificación de la PU y una ubicación de referencia asociada con la región de referencia. Por ejemplo, el MV puede ser un vector de dos dimensiones que proporciona un desplazamiento de las coordenadas en la imagen decodificada actual a coordenadas en una imagen de referencia. La unidad de estimación de movimiento puede emitir el índice de referencia y el MV como la información de movimiento de la PU. La unidad de compensación de movimiento de la unidad de procesamiento de inter-predicción 120 puede generar los bloques de muestra de predicción de la PU a partir de muestras reales o interpoladas en la ubicación de referencia indicada por el vector de movimiento de la PU.

[00109] Si una PU se encuentra en un corte B, la unidad de estimación de movimiento puede llevar a cabo una predicción unidireccional o predicción bidireccional para la PU. Para llevar a cabo predicción unidireccional para la PU, la unidad de estimación de movimiento puede buscar las imágenes de referencia de RefPicList0 o una segunda lista de imágenes de referencia ("RefPicList1") para una región de referencia para la PU. La unidad de estimación de movimiento puede emitir, como la información de movimiento de la PU, un índice de referencia que indica una posición en RefPicList0 o RefPicList1 de la imagen de referencia que contiene la región de referencia, un MV que indica un desplazamiento espacial entre un bloque de muestra de la PU y una ubicación de referencia asociada con la región de referencia, y uno o más indicadores de dirección de predicción que indican si la imagen de referencia está en RefPicList0 o RefPicList1. La unidad de compensación de movimiento de la unidad de procesamiento de inter-predicción 120 puede generar los bloques de muestra de predicción de la PU con base por lo menos en parte en

muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.

[00110] Para llevar a cabo una inter-predicción bidireccional para una PU, la unidad de estimación de movimiento puede buscar las imágenes de referencia en RefPicList0 para una región de referencia para la PU y también puede buscar las imágenes de referencia en RefPicList1 para otra región de referencia para la PU. La unidad de estimación de movimiento puede generar índices de imágenes de referencia que indican las posiciones en RefPicList0 y RefPicList1 de las imágenes de referencia que contienen las regiones de referencia. Además, la unidad de estimación de movimiento puede generar MV que indican los desplazamientos espaciales entre la ubicación de referencia asociada con las regiones de referencia y un bloque de muestra 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 muestra de predicción de la PU con base por lo menos en parte en muestras reales o interpoladas en la región de referencia indicada por el vector de movimiento de la PU.

[00111] De acuerdo con diversos ejemplos de esta revelación, el codificador de video 20 puede estar configurado para llevar a cabo la codificación basada en paletas. Con respecto al marco de HEVC, como un ejemplo, las técnicas de codificación basada en paletas pueden estar configuradas para ser utilizadas como un modo de la unidad de codificación (CU). En otros ejemplos, las técnicas de codificación basada en paletas pueden estar configuradas para ser utilizado como un modo de PU en el marco de HEVC. En consecuencia, todos los procesos revelados en la presente memoria que se describen (en toda esta revelación) en el contexto de un modo de CU puede, en forma adicional o alternativa, aplicarse a las PU. Sin embargo, estos ejemplos basados en HEVC no se deben considerar como una restricción o limitación de las técnicas de codificación basada en paletas descriptas en la presente memoria, ya que tales técnicas se pueden aplicar para funcionar en forma independiente o como parte de otros sistemas/estándares existentes o aún por desarrollar. En estos casos, la unidad para la codificación por paletas puede ser bloques cuadrados, bloques rectangulares, o incluso regiones de forma no rectangular.

[00112] La unidad de codificación basada en paletas 122, por ejemplo, puede llevar a cabo la codificación basada en paletas cuando se selecciona un modo de codificación basado en paletas, por ej., para una CU o PU. Por ejemplo, la unidad de codificación basada en paletas 122 puede estar configurada para generar una paleta

que tiene entradas que indican los valores de píxel, seleccionar valores de píxel en una paleta para representar valores de píxel de por lo menos algunas posiciones de un bloque de datos de video, y la señal de información que asocia por lo menos algunas de las posiciones del bloque de datos de video con entradas en la paleta correspondiente, respectivamente, a los valores de píxel seleccionados. Si bien varias funciones se describen como siendo realizadas por la unidad de codificación basada en paletas 122, algunas o todas de tales funciones pueden ser llevadas a cabo por otras unidades de procesamiento, o una combinación de diferentes unidades de procesamiento.

[00113] La unidad de codificación basada en paletas 122 puede generar elementos de sintaxis para definir una paleta para un bloque de datos de video. Algunos elementos de sintaxis representativos que la unidad de codificación basada en paletas 122 puede generar para definir una paleta actual para un bloque actual de datos de video incluyen, pero no se limitan a, un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de la paleta actual (por ej., **palette_transpose_flag**), uno o más elementos de sintaxis relacionados con el parámetro de cuantificación delta (QP) (por 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 de sintaxis relacionados con compensaciones de QP croma para el bloque actual de datos de video, uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual (por ej., **palette_predictor_run**), uno o más elementos de sintaxis que indican un número de entradas en la paleta actual que están señalizadas en forma explícita (por ej., **num_signalled_palette_entries**), uno o más elementos de sintaxis que indican un valor de un componente en una entrada de paleta en la paleta actual (por ej., **palette_entry**), uno o más elementos de sintaxis que indican si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada (por ej., **palette_escape_val_present_flag**), uno o más elementos de sintaxis que indican un número de entradas en la paleta actual que están señalizadas o inferidas en forma explícita (por ej., **num_palette_indices_idc**), y uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales (por ej., **palette_index_idc**). La unidad de codificación basada en paletas 122 puede emitir los elementos de sintaxis generados que definen la paleta actual para el bloque actual a uno o más de otros componentes del codificador de video 20, tal como

la unidad de codificación de la entropía 118.

[00114] De acuerdo con ello, el codificador de video 20 puede estar configurado para codificar bloques de datos de video por el uso de los modos de código con base en la paleta de acuerdo con lo descripto en esta revelación. El codificador de video 20 puede codificar selectivamente un bloque de datos de video por el uso de un modo de codificación por paletas, o codificar un bloque de datos de video por el uso de un modo diferente, por ej., tal como un modo de codificación de HEVC inter-predictivo o intra-predictivo. El bloque de datos de video puede ser, por ejemplo, una CU o PU generada de acuerdo con un proceso de codificación de HEVC. Un codificador de video 20 puede codificar algunos bloques con los modos de codificación espacial de predicción temporal inter-predictivos o intra-predictivos y decodificar otros bloques con el modo de codificación basado en paletas.

[00115] La unidad de procesamiento de intra-predicción 126 puede generar datos de predicción para una PU por medio de la realización de intra-predicción en la PU. Los datos de predicción de la PU pueden incluir bloques de muestra de predicción de la PU y varios elementos de sintaxis. La unidad de procesamiento de intra-predicción 126 puede llevar a cabo la intra-predicción sobre las PU en cortes I, cortes P, y cortes B.

[00116] Para llevar a cabo la intra-predicción en una PU, la unidad de procesamiento de intra-predicción 126 puede utilizar varios modos de intra-predicción para generar múltiples conjuntos de datos de predicción para la PU. Para utilizar un modo de intra-predicción para generar un conjunto de datos de predicción de la PU, la unidad de procesamiento de intra-predicción 126 puede extender muestras desde bloques de muestra de las PU vecinas a través de los bloques de muestra de la PU en una dirección asociada con el modo de intra-predicción. Las PU vecinas pueden estar arriba, arriba y hacia la derecha, arriba y hacia la izquierda o hacia la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha y de arriba a abajo para las PU, CU, y CTU. La unidad de procesamiento de intra-predicción 126 puede utilizar varios números de modos de intra-predicción, por ej., 33 modos de intra-predicción direccionales. En algunos ejemplos, el número de modos de intra-predicción puede depender del tamaño de la región asociada con la PU.

[00117] La unidad de procesamiento de predicción 100 puede seleccionar los datos de predicción de las PU de una CU de entre los datos de predicción generados por la unidad de procesamiento de inter-predicción 120 para las PU o los datos de predicción generados por la unidad de procesamiento de intra-predicción 126 para las

PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos de predicción de la PU de la CU con base en métricas de tasa/distorsión de los conjuntos de datos de predicción. Los bloques de muestra de predicción de los datos de predicción seleccionados se pueden denominar en la presente memoria como los bloques de muestra de predicción seleccionados.

[00118] La unidad de generación residual 102 puede generar, con base en el bloque de codificación luma, Cb y Cr de una CU y el luma de predicción seleccionado, los bloques Cb y Cr de las PU de la CU, un bloque residual luma, Cb y Cr de la CU. Por ejemplo, la unidad de generación residual 102 puede generar los bloques residuales de la CU de manera tal que cada muestra en los bloques residuales tenga un valor igual a una diferencia entre una muestra en un bloque de codificación de la CU y una muestra correspondiente en un bloque de muestra de predicción seleccionado correspondiente de una PU de la CU.

[00119] La unidad de procesamiento de transformada 104 puede llevar a cabo un particionamiento en árbol cuádruple para dividir los bloques residuales asociados a una CU en bloques de transformada asociados a TU de la CU. Por lo tanto, una TU puede estar asociada con un bloque de transformada luma y dos bloques de transformada croma. Los tamaños y las posiciones de los bloques de transformada luma y croma de las TU de una CU pueden o no estar basados en los tamaños y posiciones de los bloques de predicción de las PU de la CU. Una estructura en árbol cuádruple conocida como "árbol cuádruple residual" (RQT) puede incluir nodos asociados con cada una de las regiones. Las TU de una CU pueden corresponder a nodos hoja del RQT.

[00120] La unidad de procesamiento de transformada 104 puede generar bloques de coeficientes de transformada para cada TU de una CU por medio de la aplicación de una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 104 puede aplicar varias transformadas a un bloque de transformada asociado con una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada discreta del coseno (DCT), una transformada direccional, o una transformada conceptualmente similar a un bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 104 no aplica transformaciones a un bloque de transformada. En tales ejemplos, el bloque de transformada puede ser tratado como un bloque de coeficiente de transformada.

[00121] 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 con algunos o todos los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de *n* bits se puede redondear hacia abajo a 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 con una TU de una CU con base en un valor de parámetro de cuantificación (QP) asociado con la CU. El codificador de video 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados con una CU por medio del ajuste del valor de QP asociado con la CU. La cuantificación puede introducir la pérdida de información, por lo tanto, los coeficientes de transformada cuantificados pueden tener menor precisión que los originales.

[00122] La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformada inversa 110 pueden aplicar cuantificación inversa y transformadas inversas a un bloque de coeficientes, respectivamente, para reconstruir un bloque residual del bloque de coeficientes. La unidad de reconstrucción 112 puede añadir el bloque residual reconstruido a muestras correspondientes de uno o más bloques de muestra de predicción generados por la unidad de procesamiento de predicción 100 para producir un bloque de transformada reconstruido asociado con una TU. Por medio de la reconstrucción de bloques de transformada para cada TU de una CU de esta manera, el codificador de video 20 puede reconstruir los bloques de codificación de la CU.

[00123] La unidad de filtro 114 puede llevar a cabo una o más operaciones de desbloqueo para reducir los artefactos de bloqueo en los bloques de codificación asociados con una CU. El búfer de imagen decodificada 116 puede almacenar los bloques de codificación reconstruidos después de la unidad de filtro 114 lleva a cabo las una o más operaciones de desbloqueo en los bloques de codificación reconstruidos. La unidad de procesamiento de inter-predicción 120 puede utilizar una imagen de referencia que contiene los bloques de codificación reconstruidos para llevar a cabo una inter-predicción en las PU de otras imágenes. Además, la unidad de procesamiento de intra-predicción 126 puede utilizar bloques de codificación reconstruidos en el búfer de imagen decodificada 116 para llevar a cabo la intra-predicción en otras PU en la misma imagen que la CU.

[00124] La unidad de codificación de la entropía 118 puede recibir datos de otros componentes funcionales del codificador de video 20. Por ejemplo, la unidad de codificación de la entropía 118 puede recibir bloques de coeficientes de la unidad de

cuantificación 106 y puede recibir elementos de sintaxis de la unidad de procesamiento de predicción 100. La unidad de codificación de la entropía 118 puede llevar a cabo una o más operaciones de codificación de la entropía de los datos para generar datos codificados por entropía. Por ejemplo, la unidad de codificación de la entropía 118 puede llevar a cabo una operación de codificación adaptativa según el contexto de longitud variable (CAVLC), una operación de CABAC, una operación de codificación de longitud de variable-a variable (V2V), una operación de codificación de aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), una operación de codificación de la entropía de Particionamiento de Intervalo de Probabilidad (PIPE), una operación de codificación Golomb-Exponencial, u otro tipo de operación de codificación de la entropía sobre los datos. El codificador de video 20 puede emitir una corriente de bits que incluye datos codificados por entropía generados por la unidad de codificación de la entropía 118. Por ejemplo, la corriente de bits puede incluir datos que representan un RQT para una CU.

[00125] De acuerdo con lo discutido con anterioridad, la unidad de codificación basada en paletas 122 puede dar salida a los elementos de sintaxis generados que definen la paleta actual para el bloque actual a la unidad de codificación de la entropía 118. La unidad de codificación de la entropía 118 puede codificar uno o más compartimientos de los elementos de sintaxis recibidos de la unidad de codificación basada en paletas 122 por el uso de CABAC con contextos y uno o más compartimientos de los elementos de sintaxis recibidos de la unidad de codificación basada en paletas 122 por el uso de CABAC sin contextos (es decir, el modo de derivación). En algunos ejemplos, la unidad de codificación de la entropía 118 puede codificar los compartimientos de los elementos de sintaxis por el uso de contextos o el modo de derivación de acuerdo con lo definido con anterioridad en la Tabla 2.

[00126] De acuerdo con lo discutido con anterioridad, puede ser deseable agrupar los compartimientos codificados de derivación juntos para aumentar el rendimiento de CABAC. En el Proyecto 3 de SCC, los compartimientos de los elementos de sintaxis **palette_predictor_run**, **num_signalled_palette_entries**, **palette_entry**, y **palette_escape_val_present_flag** se codifican por derivación y se agrupan juntos. Sin embargo, si bien los compartimientos de los elementos de sintaxis **num_palette_indices_idc**, y **palette_index_idc** también son codificados por derivación, no están agrupados con los compartimientos de los elementos de sintaxis **palette_predictor_run**, **num_signalled_palette_entries**, **palette_entry**, y **palette_escape_val_present_flag**. En cambio, en el Proyecto 3 de HEVC SCC, los

elementos de sintaxis **num_palette_indices_idc**, y **palette_index_idc** se separan de los elementos de sintaxis **palette_predictor_run**, **num_signalled_palette_entries**, **palette_entry**, y **palette_escape_val_present_flag** por medio de uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para un bloque actual de datos de video (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 de sintaxis que indica si se aplica un proceso de transposición a los índices de paleta de la paleta para el bloque actual de datos de video (es decir, **palette_transpose_flag**).

[00127] De acuerdo con una o más técnicas de esta revelación, la unidad de codificación de la entropía 118 puede codificar los elementos de sintaxis que se utilizan para definir una paleta actual de manera tal que los elementos de sintaxis que se codifican por el uso del modo de derivación sean codificados en forma consecutiva. Por ejemplo, en oposición a la separación de los compartimientos de los elementos de sintaxis **palette_predictor_run**, **num_signalled_palette_entries**, **palette_entry**, y **palette_escape_val_present_flag** y los compartimientos de los elementos de sintaxis **num_palette_indices_idc**, y **palette_index_idc**, la unidad de codificación de la entropía 118 puede codificar uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video después de un elemento de sintaxis que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de video de manera tal que los compartimientos de los elementos de sintaxis **palette_predictor_run**, **num_signalled_palette_entries**, **palette_entry**, y **palette_escape_val_present_flag**, **num_palette_indices_idc**, y **palette_index_idc** se agrupen juntos. De esta manera, el rendimiento de CABAC de la unidad de codificación de la entropía 118 se puede incrementar.

[00128] La FIG. 3 es un diagrama de bloques que ilustra un decodificador de video representativo 30 que está configurado para poner en práctica las técnicas de esta revelación. La FIG. 3 se proporciona con fines de explicación y no es limitativo en las técnicas de la manera más amplia ejemplificadas y descriptas en esta revelación. Para fines de explicación, esta revelación describe el decodificador de video 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta revelación pueden ser aplicables a otros estándares o métodos de codificación.

[00129] El decodificador de video 30 representa un ejemplo de un dispositivo

que se puede configurar para llevar a cabo las técnicas de codificación de video basadas en paletas de acuerdo con diversos ejemplos que se describen en esta revelación. Por ejemplo, el decodificador de video 30 puede estar configurado para decodificar en forma selectiva varios bloques de datos de video, tales como CU o PU en la codificación de HEVC, por el uso de ya sea codificación basada en paletas o codificación no basada en paletas. Los modos de codificación no basados en paletas se pueden referir a diversos modos de codificación temporales de inter-predicción o modos de codificación espaciales de intra-predicción, tales como los diversos modos de codificación especificados por el estándar de HEVC. El decodificador de video 30, en un ejemplo, puede estar configurado para generar una paleta que tiene entradas que indican los valores de píxel, recibir información que asocia por lo menos algunas posiciones de un bloque de datos de video con entradas en la paleta, seleccionar valores de píxel en la paleta con base en la información, y reconstruir los valores de píxel del bloque con base en los valores de píxel seleccionados.

[00130] En el ejemplo de la FIG. 3, el decodificador de video 30 incluye una unidad de decodificación de la 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 un búfer de imagen decodificada 162. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 164 y una unidad de procesamiento de intra-predicción 166. El decodificador de video 30 también incluye una unidad de decodificación basada en paletas 165 configurado para llevar a cabo varios aspectos de las técnicas de codificación basada en paletas que se describen en esta revelación. En otros ejemplos, el decodificador de video 30 puede incluir más o menos, o diferentes componentes funcionales.

[00131] En algunos ejemplos, el decodificador de video 30 puede incluir además, una memoria de datos de video 149. La memoria de datos de video 149 puede almacenar datos de video, tal como una corriente de bits de video codificado, para ser decodificados por los componentes del decodificador de video 30. Los datos de video almacenados en la memoria de datos de video 149 se pueden obtener, por ejemplo, desde el canal 16, por ej., desde una fuente de video local, tal como una cámara, a través de la comunicación de red por cable o inalámbrica de datos de video, o por medio del acceso a los medios de almacenamiento de datos físicos. La memoria de datos de video 149 puede formar un búfer de imagen codificada (CPB) que almacena datos de video codificados a partir de una corriente de bits de video

codificado. El CPB puede ser una memoria de imágenes de referencia que almacena datos de video de referencia para su uso en la decodificación de datos de video por el decodificador de video 30, por ej., en los modos de intra o inter-codificación. La memoria de datos de video 149 puede estar formada por cualquiera de una variedad de dispositivos de memoria, tales como la memoria dinámica de acceso aleatorio (DRAM), que incluyen DRAM síncrona (SDRAM), RAM magneto-resistivo (MRAM), RAM resistivo (RRAM), u otros tipos de dispositivos de memoria. La memoria de datos de video 149 y el búfer de imagen decodificada 162 pueden ser proporcionados por el mismo dispositivo de memoria o dispositivos de memoria separados. En diversos ejemplos, la memoria de datos de video 149 pueden ser en chip con otros componentes del decodificador de video 30, o fuera del chip con respecto a esos componentes.

[00132] Un búfer de imagen codificada (CPB) puede recibir y almacenar datos de video codificados (por ej., las unidades NAL) de una corriente de bits. La unidad de decodificación de la entropía 150 podrá recibir datos de video codificados (por ej., unidades NAL) del CPB y analizar las unidades NAL para decodificar los elementos de sintaxis. La unidad de decodificación de la entropía 150 puede decodificar por entropía elementos de sintaxis codificados por entropía en las unidades NAL. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de procesamiento de transformada inversa 156, la unidad de reconstrucción 158, y la unidad de filtro 160 puede generar datos de video decodificados con base en los elementos de sintaxis extraídos de la corriente de bits.

[00133] Las unidades NAL de la corriente de bits pueden incluir unidades NAL de corte codificado. Como parte de la decodificación de la corriente de bits, la unidad de decodificación de la entropía 150 puede extraer y decodificar los elementos de sintaxis a partir de entropía de unidades NAL de corte codificado. Cada uno de los cortes codificados puede incluir una cabecera de corte y datos de corte. La cabecera de corte puede contener elementos de sintaxis que pertenecen a un corte. Los elementos de sintaxis en la cabecera de corte pueden incluir un elemento de sintaxis que identifica un PPS asociado con una imagen que contiene el corte.

[00134] Además de la decodificación de elementos de sintaxis desde la corriente de bits, el decodificador de video 30 puede llevar a cabo una operación de reconstrucción en una CU sin particiones. Para llevar a cabo la operación de reconstrucción en una CU sin particiones, el decodificador de video 30 puede llevar a cabo una operación de reconstrucción en cada TU de la CU. Por medio de la

realización de la operación de reconstrucción para cada TU de la CU, el decodificador de video 30 puede reconstruir bloques residuales de la CU.

[00135] Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede cuantificar en forma inversa, es decir, descuantificar, bloques de coeficientes asociados con la TU. La unidad de cuantificación inversa 154 puede utilizar un valor de QP asociado con la CU de la TU para determinar un grado de cuantificación y, así mismo, un grado de cuantificación inversa para la unidad de cuantificación inversa 154 para aplicar. Es decir, la relación de compresión, es decir, la relación entre el número de bits utilizados para representar una secuencia original y la secuencia comprimida, se pueden controlar por medio del ajuste del valor del QP utilizado cuando se cuantifican coeficientes de transformada. La relación de compresión también puede depender del método de codificación de la entropía empleado.

[00136] Después de que la unidad de cuantificación inversa 154 cuantifica en forma inversa un bloque de coeficientes, la unidad de procesamiento de transformada inversa 156 puede aplicar una o más transformadas inversas al bloque de coeficientes con el fin de generar un bloque residual asociado con la TU. Por ejemplo, la unidad de procesamiento de transformada inversa 156 puede aplicar una DCT inversa, una transformada inversa de número entero, una transformada inversa de Karhunen-Loeve (KLT), una transformada de rotación inversa, una transformada de dirección inversa, u otra transformada inversa al bloque de coeficientes.

[00137] Si una PU se codifica por el uso de intra-predicción, la unidad de procesamiento de intra-predicción 166 puede llevar a cabo la intra-predicción para generar los bloques de predicción de la PU. La unidad de procesamiento de intra-predicción 166 puede utilizar un modo de intra-predicción para generar los bloques de predicción luma, Cb y Cr de la PU con base en los bloques de predicción de PU espacialmente vecinas. La unidad de procesamiento de intra-predicción 166 puede determinar el modo de intra-predicción para la PU con base en uno o más elementos de sintaxis de la corriente de bits decodificada.

[00138] 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) con base en los elementos de sintaxis extraídos de la corriente de bits. Además, si una PU se codifica por el uso de inter-predicción, la unidad de decodificación de la entropía 150 puede extraer la información de movimiento para la PU. La unidad de compensación de movimiento 164 puede

determinar, con base en la información de movimiento de la PU, una o más regiones de referencia para la PU. La unidad de compensación de movimiento 164 puede generar, con base en muestras de bloques en los uno o más bloques de referencia para la PU, los bloques de predicción luma, Cb y Cr para la PU.

[00139] La unidad de reconstrucción 158 puede utilizar los bloques de transformada luma, Cb y Cr asociados con las TU de una CU y los bloques de predicción luma, Cb y Cr de las PU de la CU, es decir, cualquiera de los datos de intra-predicción o los datos de inter-predicción, de acuerdo con el caso, para reconstruir los bloques de codificación luma, Cb y Cr de la CU. Por ejemplo, la unidad de reconstrucción 158 puede añadir muestras de los bloques de transformada luma, Cb y Cr a muestras correspondientes de los bloques de predicción luma, Cb y Cr para reconstruir los bloques de codificación luma, Cb y Cr de la CU.

[00140] La unidad de filtro 160 puede llevar a cabo una operación de desbloqueo para reducir los artefactos de bloqueo asociados con los bloques de codificación luma, Cb y Cr de la CU. El decodificador de video 30 puede almacenar los bloques de codificación luma, Cb y Cr de la CU en el búfer de imagen decodificada 162. El búfer de imagen decodificada 162 puede proporcionar imágenes de referencia para la compensación de movimiento subsecuente, intra-predicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el decodificador de video 30 puede llevar a cabo, con base en los bloques luma, Cb y Cr en el búfer de imagen decodificada 162, las operaciones de intra-predicción o inter-predicción en las PU de otras CU. De esta manera, el decodificador de video 30 puede extraer, de la corriente de bits, niveles de coeficientes de transformada del bloque de coeficientes significativo de luma, cuantificar en forma inversa los niveles de coeficientes de transformada, aplicar una transformada para los niveles de coeficientes de transformada para generar un bloque de transformada, generar, con base por lo menos en parte en el bloque de transformada, un bloque de codificación, y dar salida al bloque de codificación para la visualización.

[00141] De acuerdo con diversos ejemplos de esta revelación, el decodificador de video 30 puede estar configurado para llevar a cabo la codificación basada en paletas. La unidad de decodificación basada en paletas 165, por ejemplo, puede llevar a cabo la decodificación basada en paletas cuando se selecciona un modo de decodificación basada en paletas, por ej., para una CU o PU. Por ejemplo, la unidad de decodificación basada en paletas 165 se puede configurar para generar una paleta que tiene entradas que indican los valores de píxel, recibir información que asocia por

lo menos algunas posiciones de un bloque de datos de video con entradas en la paleta, seleccionar valores de píxel de la paleta con base en la información, y reconstruir los valores de píxel del bloque con base en los valores de píxel seleccionados. Si bien varias funciones se describen como siendo realizadas por la unidad de decodificación basada en paletas 165, algunas o todas tales funciones pueden ser llevadas a cabo por otras unidades de procesamiento, o una combinación de diferentes unidades de procesamiento.

[00142] La unidad de decodificación basada en paletas 165 puede recibir información del modo de codificación por paletas, y llevar a cabo las operaciones anteriores, cuando la información del modo de codificación por paletas indica que el modo de codificación por paletas se aplica al bloque. Cuando la información del modo de codificación por paletas indica que el modo de codificación por paletas no se aplica al bloque, o cuando otra información del modo indica el uso de un modo diferente, la unidad de procesamiento de predicción 152 decodifica el bloque de datos de video por el uso de un modo de codificación no basado en paletas, por ej., tal como HEVC inter-predictiva por el uso de la unidad de compensación de movimiento 164 o modo de codificación intra-predictivo por el uso de la unidad de procesamiento de intra-predicción 166, cuando la información del modo de codificación por paletas indica que el modo de codificación por paletas no se aplica al bloque. El bloque de datos de video puede ser, por ejemplo, una CU o PU generada de acuerdo con un proceso de codificación de HEVC. El decodificador de video 30 puede decodificar algunos bloques con los modos de codificación espacial de predicción temporal de inter-predicción o intra-predicción y decodificar otros bloques con el modo de codificación basado en paletas. El modo de codificación basado en paletas puede comprender uno de una pluralidad de diferentes modos de codificación basados en paletas, o puede ser un modo de codificación basado en paletas único.

[00143] La información del modo de codificación por paletas recibida por la unidad de decodificación basada en paletas 165 puede comprender un elemento de sintaxis en modo de paleta, tal como un marcador. Un primer valor del elemento de sintaxis en modo de paleta indica que el modo de codificación por paletas se aplica al bloque y un segundo valor del elemento de sintaxis en modo de paleta indica que el modo de codificación por paletas no se aplica al bloque de datos de video. La unidad de decodificación basada en paletas 165 puede recibir la información del modo de codificación por paletas en uno o más de un nivel de unidad de predicción, un nivel de unidad de codificación, un nivel de macro, o un nivel de imagen, o puede recibir la

información del modo de codificación por paletas en por lo menos uno de conjunto de parámetros de imagen (PPS), un conjunto de parámetros de secuencia (SPS) o un conjunto de parámetros de video (VPS).

[00144] En algunos ejemplos, la unidad de decodificación basada en paletas 165 puede inferir la información del modo de codificación por paletas con base en uno o más de un tamaño de bloque de codificación, un tipo de trama, un espacio de color, un componente de color, un tamaño de trama, una velocidad de fotogramas, una ID de la capa en la codificación de video escalable o una ID de vista en la codificación de múltiples vistas asociada con el bloque de datos de video.

[00145] La unidad de decodificación basada en paletas 165 también puede estar configurada para recibir información que define por lo menos algunas de las entradas en la paleta con datos de video, y generar la paleta con base, por lo menos en parte, en la información recibida. El tamaño de la paleta puede ser fijo o variable. En algunos casos, el tamaño de la paleta es variable y se puede ajustar con base en la información señalizada con los datos de video. La información señalizada puede especificar si una entrada en la paleta es una última entrada de paleta. También, en algunos casos, la paleta puede tener un tamaño máximo.

[00146] La paleta puede ser una sola paleta que incluye las entradas que indican los valores de píxel para un componente luma y los componentes croma del bloque. En este caso, cada entrada en la paleta es una triple entrada que indica los valores de píxel para el componente luma y dos componentes croma. Como alternativa, la paleta comprende una paleta luma que incluye entradas que indican valores de píxel de un componente luma del bloque, y las paletas croma que incluyen entradas que indican los valores de píxel para los respectivos componentes croma del bloque.

[00147] En algunos ejemplos, la unidad de decodificación basada en paletas 165 puede generar la paleta por medio de la predicción de las entradas de paleta con base en los datos previamente procesados. Los datos previamente procesados pueden incluir paletas, o informaciones de paletas, para los bloques vecinos previamente decodificados. La unidad de decodificación basada en paletas 165 puede recibir un elemento de sintaxis de predicción que indica si las entradas en la paleta se han de predecir. El elemento de sintaxis de predicción puede incluir una pluralidad de elementos de sintaxis de predicción que indican, respectivamente, si las entradas en paletas para los componentes luma y croma se han de predecir.

[00148] La unidad de decodificación basada en paletas 165 puede, en algunos

ejemplos, predecir por lo menos algunas de las entradas de paleta con base en las entradas en una paleta para un bloque vecino izquierdo o un bloque vecino superior en un corte o una imagen. En este caso, las entradas de paleta que se predicen con base en entradas en cualquiera de una paleta para el bloque vecino izquierdo o el bloque vecino superior se pueden predecir por medio de una unidad de decodificación basada en paletas 165 con base en un elemento de sintaxis que indica la selección del bloque vecino izquierdo o el bloque vecino superior para la predicción. El elemento de sintaxis puede ser un marcador que tiene un valor que indica la selección del bloque vecino izquierdo o el bloque vecino superior para la predicción.

[00149] En algunos ejemplos, la unidad de decodificación basada en paletas 165 puede recibir uno o más elementos de sintaxis de predicción que indican si por lo menos algunas de las entradas seleccionadas en la paleta, sobre una base de entrada por entrada, se han de predecir, y generar las entradas en consecuencia. La unidad de decodificación basada en paletas 165 puede predecir algunas de las entradas y recibir información por medio de la especificación directa de otras entradas de paleta.

[00150] La información recibida por la unidad de decodificación basada en paletas 165, que asocia por lo menos algunas posiciones de un bloque de datos de video con entradas en la paleta, puede comprender información del mapa que incluye valores de índice de paleta para por lo menos algunas de las posiciones en el bloque, en el que cada uno de los valores de índice de paleta corresponde a una de las entradas de paleta. La información del mapa puede incluir uno o más elementos de sintaxis de ejecución que indican cada uno un número de posiciones consecutivas en el bloque que tiene el mismo valor de índice de paleta.

[00151] En algunos ejemplos, la unidad de decodificación basada en paletas 165 puede recibir información que indica la copia de línea por medio de la cual las entradas de paleta para una línea de posiciones en el bloque se copian de entradas de paleta de un nuevo tramo de posiciones en el bloque. La unidad de decodificación basada en paletas 165 puede utilizar esta información para llevar a cabo la copia de línea para determinar las entradas de paleta para diferentes posiciones de un bloque. La línea de las posiciones puede comprender una fila, una porción de una fila, una columna o una porción de una columna de las posiciones del bloque.

[00152] La unidad de decodificación basada en paletas 165 puede generar la paleta, en parte, por medio de la recepción de valores de píxel para una o más posiciones del bloque, y la adición de los valores de píxel a las entradas de paleta para generar dinámicamente por lo menos una porción de la paleta sobre la marcha. La

adición de los valores de píxel puede comprender la adición de los valores de píxel a una paleta inicial que comprende un conjunto inicial de entradas, o para una paleta vacía que no incluye un conjunto inicial de entradas. En algunos ejemplos, la adición comprende la adición de los valores de píxel para agregar nuevas entradas a una paleta inicial que comprende un conjunto inicial de entradas o rellenar las entradas existentes en la paleta inicial, o la sustitución o cambio de los valores de píxel de entradas en la paleta inicial.

[00153] 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 de un valor de píxel real de la posición en el bloque, de manera tal que el proceso de decodificación es con pérdida. Por ejemplo, el mismo valor de píxel se puede seleccionar desde la paleta para dos posiciones diferentes que tienen diferentes valores de píxeles reales.

[00154] De acuerdo con lo discutido con anterioridad, la unidad de decodificación basada en paletas 165 puede recibir información que define una paleta para un bloque actual de datos de video. Por ejemplo, la unidad de decodificación basada en paletas 165 puede recibir una pluralidad de elementos de sintaxis de la unidad de decodificación de la entropía 150. En algunos ejemplos, la unidad de decodificación de la entropía 150 puede decodificar la pluralidad de elementos de sintaxis de una corriente de bits de video codificado de acuerdo con una tabla de sintaxis. Como un ejemplo, la unidad de decodificación de la entropía 150 puede decodificar la pluralidad de elementos de sintaxis de una corriente de bits de video codificado de acuerdo con la tabla de sintaxis de paleta del Proyecto 3 de HEVC SCC, que se reproduce con anterioridad en la Tabla 1. Sin embargo, de acuerdo con lo discutido con anterioridad, la disposición de elementos de sintaxis en el Proyecto 3 de HEVC SCC pueden no ser óptimos. En particular, la disposición de los elementos de sintaxis en el Proyecto 3 de HEVC SCC no maximiza el número de elementos de sintaxis codificados por modo de derivación que están agrupados juntos, lo que puede disminuir el rendimiento de CABAC.

[00155] De acuerdo con una o más técnicas de esta revelación, la unidad de decodificación de la entropía 150 puede decodificar los elementos de sintaxis que se utilizan para definir una paleta actual de manera tal que los elementos de sintaxis adicionales codificados por el modo de derivación se agrupan juntos. Por ejemplo, en oposición a la separación de los compartimientos de los elementos de sintaxis **palette_predictor_run**, **num_signalled_palette_entries**, **palette_entry**, y

palette_escape_val_present_flag y los compartimientos de los elementos de sintaxis **num_palette_indices_idc**, y **palette_index_idc**, la unidad de decodificación de la entropía 150 puede decodificar uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video después de un elemento de sintaxis que indica si un proceso de transposición se aplica a los índices de paleta de la paleta para el bloque actual de datos de video de manera tal que los compartimientos de los elementos de sintaxis **palette_predictor_run**, **num_signalled_palette_entries**, **palette_entry**, y **palette_escape_val_present_flag**, **num_palette_indices_idc**, y **palette_index_idc** se agrupen juntos. Como un ejemplo, la unidad de decodificación de la entropía 150 puede decodificar los elementos de sintaxis que se utilizan para definir la paleta actual en el orden mostrado con anterioridad en la Tabla 4. Como otro ejemplo, la unidad de decodificación de la entropía 150 puede decodificar los elementos de sintaxis que se utilizan para definir la paleta actual en el orden mostrado con anterioridad en la Tabla 5. De esta manera, el rendimiento de CABAC de la unidad de decodificación de la entropía 150 puede ser aumentado.

[00156] La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de determinación de una paleta para la codificación de datos de video, de acuerdo con las técnicas de esta revelación. El ejemplo de la FIG. 4 incluye una imagen 178 que tiene una primera unidad de codificación (CU) 180 que está asociada con las primeras paletas 184 y una segunda CU 188 que está asociada con las segundas paletas 192. De acuerdo con lo descripto en mayor detalle a continuación y de acuerdo con las técnicas de esta revelación, las segundas paletas 192 están basadas en las primeras paletas 184. La imagen 178 también incluye el bloque 196 que está codificado con un modo de codificación de intra-predicción y el bloque 200 que está codificado con un modo de codificación de inter-predicción.

[00157] Las técnicas de la FIG. 4 se describen en el contexto del codificador de video 20 (FIG. 1 y FIG. 2) y el decodificador de video 30 (FIG. 1 y FIG. 3) y con respecto al estándar de HEVC para fines de explicación. Sin embargo, se debe entender que las técnicas de esta revelación no se limitan de esta manera, y pueden ser aplicadas por otros procesadores de codificación de video y/o dispositivos en otros procesos de codificación de video y/o estándares.

[00158] Por lo general, una paleta se refiere a un número de valores de píxel que son dominantes y/o representativos de una CU que actualmente se está codificando, tal como la CU 188 en el ejemplo de la FIG. 4. Las primeras paletas 184 y

las segundas paletas 192 se muestran como que incluyen varias paletas. En algunos ejemplos, un codificador de video (tal como el codificador de video 20 o el decodificador de video 30) puede codificar paletas por separado para cada componente de color de una CU. Por ejemplo, el codificador de video 20 puede codificar una paleta para un componente luma (Y) de una CU, otra paleta para un componente croma (U) de la CU, y aún otra paleta para el componente croma (V) de la CU. En este ejemplo, las entradas de paleta Y pueden representar valores de Y de píxeles de la CU, las entradas de paleta T pueden representar valores de U de píxeles de la CU, y las entradas de paleta V pueden representar valores de V de píxeles de la CU. En otro ejemplo, el codificador de video 20 puede codificar una paleta para un componente luma (Y) de una CU, y otra paleta de dos componentes (U, V) de la CU. En este ejemplo, las entradas de paleta Y pueden representar valores de Y de píxeles de la CU, y las entradas de paleta de U-V pueden representar pares de valores U-V de píxeles de la CU.

[00159] En otros ejemplos, el codificador de video 20 puede codificar una única paleta para todos los componentes de color de una CU. En este ejemplo, el codificador de video 20 puede codificar una paleta que tiene una entrada i-ésima que es un valor triple, que incluyen Yi, Ui, y Vi. En este caso, la paleta incluye valores para cada uno de los componentes de los píxeles. De acuerdo con ello, la representación de las paletas 184 y 192 como un conjunto de paletas que tienen múltiples paletas individuales es meramente un ejemplo y no pretende ser limitante.

[00160] En el ejemplo de la FIG. 4, las primeras paletas 184 incluye tres entradas 202 a 206 que tienen valor de índice de entrada 1, valor de índice de entrada 2, y el valor de índice de entrada 3, respectivamente. Las entradas 202 a 206 relacionan los valores de índice con los valores de píxel que incluyen el valor de píxel A, el valor de píxel B, y el valor de píxel C, respectivamente. De acuerdo con lo descripto en la presente memoria, en lugar de la codificación de los valores de píxel reales de la primera CU 180, un codificador de video (tal como el codificador de video 20 o el decodificador de video 30) pueden utilizar la codificación basada en paletas para codificar los píxeles del bloque por el uso de los índices 1 a 3. Es decir, para cada posición de píxel de la primera CU 180, el codificador de video 20 puede codificar un valor de índice para el píxel, donde el valor de índice se asocia con un valor de píxel en una o más de las primeras paletas 184. El decodificador de video 30 puede obtener los valores de índice de una corriente de bits y reconstruir los valores de píxel por el uso de los valores de índice y una o más de las primeras paletas 184. De este modo,

las primeras paletas 184 son transmitidas por el codificador de video 20 en una corriente de bits de datos de video codificada para su uso por el decodificador de video 30 en la decodificación basada en paletas. Por lo general, una o más paletas pueden ser transmitidas por cada CU o pueden ser compartidas entre diferentes CU.

[00161] El codificador de video 20 y el decodificador de video 30 puede determinar las segundas paletas 192 con base en las primeras paletas 184. Por ejemplo, el codificador de video 20 puede codificar un pred_palette_flag para cada CU (que incluyen, como un ejemplo, una segunda CU 188) para indicar si la paleta para la CU se predice a partir de una o más paletas asociadas con una o más otras CU, tal como CU vecinas (en forma espacial o con base en el orden de escaneado) o las muestras más frecuentes de un vecino de causalidad. Por ejemplo, cuando el valor de un indicador de este tipo es igual a uno, el decodificador de video 30 puede determinar que las segundas paletas 192 para la segunda CU 188 se predice a partir de una o más paletas ya decodificadas y por lo tanto no hay nuevas paletas para la segunda CU 188 que están incluidas en una corriente de bits que contiene el pred_palette_flag. Cuando un marcador tal es igual a cero, el decodificador de video 30 puede determinar que la paleta 192 de 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 ej., tres marcadores, uno para Y, uno para U, y otro para V, para una CU en el video YUV). En otros ejemplos, una única pred_palette_flag se puede codificar para todos los componentes de color de una CU.

[00162] En el ejemplo anterior, el pred_palette_flag está señalizado por CU para indicar si cualquiera de las entradas de paleta para el bloque actual se predice. En algunos ejemplos, uno o más elementos de sintaxis se pueden señalizar en una base por entrada. Es decir, un marcador se puede señalizar para cada entrada de un predictor de paleta para indicar si esa entrada está presente en la paleta actual. De acuerdo con lo indicado con anterioridad, si una entrada de paleta no está predicho, la paleta se puede señalizar en forma explícita.

[00163] Cuando se determinan las segundas paletas 192 en relación con las primeras paletas 184 (por ej., pred_palette_flag es igual a uno), el codificador de video 20 y/o el decodificador de video 30 pueden localizar uno o más bloques de los cual las paletas de predicción, en este ejemplo las primeras paletas 184, se determinan. Las paletas de predicción pueden estar asociadas con una o más CU vecinas de la CU actualmente siendo codificadas (por ej., tal como las CU vecinas (espacial o con base

en el orden de escaneado) o las muestras más frecuentes de un vecino de causalidad), es decir, la segunda CU 188. Las paletas de las una o más CU vecinas pueden estar asociadas con una paleta de predicción. En algunos ejemplos, tales como el ejemplo ilustrado en la FIG. 4, el codificador de video 20 y/o el decodificador de video 30 pueden localizar una CU vecina izquierda, una primera CU 180, cuando se determina una paleta predictiva para la segunda CU 188. En otros ejemplos, el codificador de video 20 y/o el decodificador de video 30 pueden localizar una o más CU en otras posiciones con respecto a la segunda CU 188, tal como una CU superior, la CU 196.

[00164] El codificador de video 20 y/o el decodificador de video 30 pueden determinar una CU para la predicción por paletas basada en una jerarquía. Por ejemplo, el codificador de video 20 y/o el decodificador de video 30 pueden identificar inicialmente la CU vecina izquierda, la primera CU 180, para la predicción por paletas. Si la CU vecina izquierda no está disponible para la predicción (por ej., la CU vecina izquierda está codificada con un modo distinto de un modo de codificación basado en paletas, tal como un modo de inter-predicción o el modo de intra-predicción, o se encuentra ubicado en el borde más a la izquierda de una imagen o corte) del codificador de video 20 y/o el decodificador de video 30 pueden identificar la CU vecina superior, la CU 196. El codificador de video 20 y/o el decodificador de video 30 pueden continuar la búsqueda de una CU disponible de acuerdo con un orden predeterminado de ubicaciones hasta la localización de una CU que tiene una paleta disponible para la predicción por paletas. En algunos ejemplos, el codificador de video 20 y/o el decodificador de video 30 pueden determinar una paleta de predicción con base en varios bloques y/o muestras reconstruidas de un bloque vecino.

[00165] Si bien el ejemplo de la FIG. 4 ilustra las primeras paletas 184 como paletas de predicción a partir de una única CU, la primera CU 180, en otros ejemplos, el codificador de video 20 y/o el decodificador de video 30 pueden localizar paletas de predicción a partir de una combinación de la CU vecina. Por ejemplo, el codificador de video 20 y/o el decodificador de video 30 pueden aplicar una o más fórmulas, funciones, estándares o similares para generar una paleta basada en paletas de uno o de una combinación de una pluralidad de CU vecinas.

[00166] En otros ejemplos, el codificador de video 20 y/o el decodificador de video 30 pueden construir una lista de candidatos que incluye un número de candidatos potenciales para la predicción por paletas. Un proceso de poda se puede aplicar tanto en el codificador de video 20 como el decodificador de video 30 para

eliminar los candidatos duplicados en la lista. En tales ejemplos, el codificador de video 20 puede codificar un índice de la lista de candidatos para indicar la CU candidata en la lista de la que se selecciona la CU actual utilizada para la predicción por paletas (por ej., copia la paleta). El decodificador de video 30 puede construir la lista de candidatos de la misma manera, decodificar el índice, y utilizar el índice decodificado para seleccionar la paleta de la CU correspondiente para su uso con la CU actual.

[00167] En un ejemplo con fines de ilustración, el codificador de video 20 y el decodificador de video 30 pueden construir una lista de candidatos que incluye una CU que se coloca por encima de la CU actualmente siendo codificada y una CU que se coloca a la izquierda de la CU que actualmente se está codificando. En este ejemplo, el codificador de video 20 puede codificar uno o más elementos de sintaxis para indicar la selección de candidato. Por ejemplo, el codificador de video 20 puede codificar un marcador que tiene un valor de cero para indicar que la paleta para la CU actual se copia de la CU posicionada a la izquierda de la CU actual. El codificador de video 20 puede codificar el marcador que tiene un valor de uno para indicar que la paleta para la CU actual se copia de la CU posicionada por encima de la CU actual. El decodificador de video 30 decodifica el marcador y selecciona la CU apropiada para la predicción por paletas.

[00168] En todavía otros ejemplos, el codificador de video 20 y/o el decodificador de video 30 determinar la paleta para la CU actualmente codificada con base en la frecuencia con la que los valores de muestra incluidos en una o más otras paletas ocurren en una o más CU vecinas. Por ejemplo, el codificador de video 20 y/o el decodificador de video 30 pueden llevar a cabo un monitoreo de los colores asociados a los valores de índice de uso más frecuente durante la codificación de un número predeterminado de CU. El codificador de video 20 y/o el decodificador de video 30 pueden incluir los colores más utilizados en la paleta para la CU que actualmente se está codificando.

[00169] En algunos ejemplos, el codificador de video 20 y/o el decodificador de video 30 pueden llevar a cabo la predicción por paletas basada en entradas. Por ejemplo, el codificador de video 20 puede codificar uno o más elementos de sintaxis, tales como uno o más marcadores, para cada entrada de una paleta de predicción que indica si las respectivas entradas de paleta de predicción se vuelven a utilizar en la paleta actual (por ej., si los valores de píxel en una paleta de otra CU son reutilizados por la paleta actual). En este ejemplo, el codificador de video 20 puede codificar un

marcador que tiene un valor igual a uno para una entrada dada cuando la entrada es un valor predicho a partir de una paleta de predicción (por ej., una entrada correspondiente de una paleta asociada con una CU vecina). El codificador de video 20 puede codificar un marcador que tiene un valor igual a cero para una entrada en 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 video 20 también puede codificar datos adicionales que indican el valor de la entrada de paleta no predicha.

[00170] En el ejemplo de la FIG. 4, las segunda paletas 192 incluyen cuatro entradas 208 a 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 el índice de entrada 4, respectivamente. Las entradas 208 a 214 relacionan los valores de índice de 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 video 20 y/o el decodificador de video 30 pueden utilizar cualquiera de las técnicas descriptas con anterioridad para localizar la primera CU 180 para fines de predicción por paletas y copiar las entradas 1 a 3 de las primeras paletas 184 a las entradas 1 a 3 de las segundas paletas 192 para la codificación de la segunda CU 188. De esta manera, el codificador de video 20 y/o el decodificador de video 30 pueden determinar las segundas paletas 192 con base en las primeras paletas 184. Además, el codificador de video 20 y/o el decodificador de video 30 pueden codificar los datos de entrada 4 para ser incluidos con las segundas paletas 192. Dicha información puede incluir el número de entradas de paleta no pronosticado a partir de una paleta de predicción y los valores de píxel correspondientes a dichas entradas de paleta.

[00171] En algunos ejemplos, de acuerdo con los aspectos de esta revelación, uno o más elementos de sintaxis pueden indicar si las paletas, tales como las segundas paletas 192, se predice por completo de una paleta de predicción (que se muestra en la FIG. 4 como las primeras paletas 184, pero que puede estar compuesto por las entradas de uno o más bloques) o si las entradas particulares de las segundas paletas 192 está predichas. Por ejemplo, un elemento de sintaxis inicial puede indicar si se predicen todas las entradas. Si el elemento de sintaxis inicial indica que no todas las entradas se predicen (por ej., un marcador que tiene un valor de 0), uno o más elementos de sintaxis pueden indicar que las entradas de las segundas paletas 192 se predicen a partir de la paleta de predicción.

[00172] De acuerdo con algunos aspectos de esta revelación, cierta información asociada con la predicción por paletas se puede deducir de una o más características

de los datos que están siendo codificados. Es decir, en lugar de que el codificador de video 20 codifique los elementos de sintaxis (y el decodificador de video 30 decodifique tales elementos de sintaxis), el codificador de video 20 y el decodificador de video 30 puede llevar a cabo la predicción por paletas basada en una o más características de los datos que están siendo codificados.

[00173] La FIG. 5 es un diagrama conceptual que ilustra un ejemplo de la determinación de los índices a una paleta para un bloque de píxeles, de acuerdo con las técnicas de esta revelación. Por ejemplo, la FIG. 5 incluye un mapa 240 de valores de índice (valores 1, 2, y 3) que relacionan posiciones respectivas de píxeles asociados con los valores de índice a una entrada de paletas 244. Las paletas 244 se pueden determinar de una manera similar a las primeras paletas 184 y las segundas paletas 192 descriptas con anterioridad con respecto a la FIG. 4.

[00174] Una vez más, las técnicas de la FIG. 5 se describen en el contexto del codificador de video 20 (FIG. 1 y FIG. 2) y el decodificador de video 30 (FIG. 1 y FIG. 3) y con respecto al estándar de codificación de video de HEVC para fines de explicación. Sin embargo, se debe entender que las técnicas de esta revelación no se limitan de esta manera, y se pueden aplicar por medio de otros procesadores de codificación de video y/o dispositivos en otros procesos de codificación de video y/o estándares.

[00175] Si bien el mapa 240 se ilustra en el ejemplo de la FIG. 5 como que incluye un valor de índice para cada posición de píxel, se debe entender que en otros ejemplos, no todas las posiciones de píxel pueden estar asociadas con un valor de índice que relaciona el valor de píxel con una entrada de paletas 244. Esto es, de acuerdo con lo señalado con anterioridad, en algunos ejemplos, el codificador de video 20 puede codificar (y el decodificador de video 30 pueden obtener, a partir de una corriente de bits codificada) la indicación de un valor real de píxeles (o su versión cuantificada) para una posición en el mapa 240 si el valor de píxel no está incluido en las paletas 244.

[00176] En algunos ejemplos, el codificador de video 20 y el decodificador de video 30 pueden estar configurados para codificar un mapa adicional que indica qué posiciones de píxel se asocian con valores de índice. Por ejemplo, se supone que la entrada (i, j) en el mapa corresponde a la posición (i, j) de una CU. El codificador de video 20 puede codificar uno o más elementos de sintaxis para cada entrada del mapa (es decir, cada posición de píxel) que indica si la entrada tiene un valor de índice asociado. Por ejemplo, el codificador de video 20 puede codificar un marcador que

tiene un valor de uno para indicar que el valor de píxel en la posición (i, j) en la CU es uno de los valores en las paletas 244. El codificador de video 20 puede, en tal ejemplo, también codificar un índice de paleta (que se muestra en el ejemplo de la FIG. 5 como los valores 1 a 3) para indicar ese valor de píxel en la paleta y para permitir al decodificador de video reconstruir el valor de píxel. En los casos en los que las paletas 244 incluyen una sola entrada y el valor de píxel asociado, el codificador de video 20 puede omitir la señalización del valor de índice. El codificador de video 20 puede codificar el marcador para tener un valor de cero para indicar que el valor de píxel en la posición (i, j) en la CU no es uno de los valores de paletas 244. En este ejemplo, el codificador de video 20 también puede codificar una indicación del valor de píxel para el uso por parte del decodificador de video 30 en la reconstrucción del valor de píxel. En algunos casos, el valor de píxel se puede codificar de una manera con pérdida.

[00177] El valor de un píxel en una posición de una CU puede proporcionar una indicación de los valores de uno o más de otros píxeles en otras posiciones de la CU. Por ejemplo, puede haber una probabilidad relativamente alta de que posiciones vecinas de píxel de una CU tengan el mismo valor de píxel o se pueden correlacionar con el mismo valor de índice (en el caso de la codificación con pérdidas, en la que más de un valor de píxel se puede mapear a un único valor de índice).

[00178] De acuerdo con ello, el codificador de video 20 puede codificar uno o más elementos de sintaxis que indican un número de píxeles consecutivos o valores de índice en un orden de exploración dado que tienen el mismo valor de píxel o valor de índice. De acuerdo con lo señalado con anterioridad, la cadena de valores de píxel o de valores de índice similar se puede denominar en la presente memoria como una ejecución. En un ejemplo con fines de ilustración, si dos píxeles consecutivos o índices en un orden de exploración dado tienen valores diferentes, la ejecución es igual a cero. Si dos píxeles consecutivos o índices en un orden de exploración dado tienen el mismo valor, pero el tercer píxel o índice en el orden de exploración tiene un valor diferente, la ejecución es igual a uno. Para tres índices o píxeles consecutivos con el mismo valor, la ejecución es de dos, y así sucesivamente. El decodificador de video 30 puede obtener los elementos de sintaxis que indican una ejecución de una corriente de bits codificada y utilizar los datos para determinar el número de posiciones consecutivas que tienen el mismo píxel o valor de índice.

[00179] El número de índices que se pueden incluir en una ejecución se puede ver afectado por el orden de exploración. Por ejemplo, si se considera una exploración de trama de líneas 266, 268, y 270 del mapa 240. Suponiendo una dirección de

exploración horizontal, de izquierda a derecha (tal como un orden de exploración de trama), la fila 266 incluye tres valores de índice "1", dos valores de índice "2", y tres valores de índice "3". La fila 268 incluye cinco valores de índice "1" y tres de valores índice "3”. En este ejemplo, para la fila 266, el codificador de video 20 puede codificar los elementos de sintaxis que indica que el primer valor de la fila 266 (el valor más a la izquierda de la fila) es 1 con una ejecución 2, seguido por un valor de índice 2 con una ejecución 1, seguido por un valor de índice 3 con una ejecución 2. Tras la exploración de trama, el codificador de video 20 puede entonces comenzar la fila de codificación 268 con el valor más a la izquierda. Por ejemplo, el codificador de video 20 puede codificar los elementos de sintaxis que indican que el primer valor de la fila 268 es 1 con una ejecución 4, seguido por un valor de índice 3 con una ejecución 2. El codificador de video 20 puede proceder de la misma manera con la línea 270.

[00180] Por lo tanto, en el orden de exploración de trama, el primer índice de una línea actual se puede escanear directamente después del último índice de una línea anterior. Sin embargo, en algunos ejemplos, puede que no sea deseable escanear los índices en un orden de exploración de trama. Por ejemplo, puede que no sea deseable escanear los índices en un orden de exploración de trama, donde una primera línea de un bloque de datos de video (por ej., la fila 266) incluye un primer píxel adyacente a un primer borde del bloque de datos de video (por ej., el píxel más a la izquierda de la fila 266, que tiene un valor de índice 1) y un último píxel adyacente a un segundo borde del bloque de datos de video (por ej., el píxel más a la derecha de la fila 266, que tiene un valor de índice 3), una segunda línea del bloque de datos de video (por ej., la fila 268) incluye una primera píxel adyacente al primer borde del bloque de datos de video (por ej., el píxel más a la izquierda de la fila 268, que tiene un valor de índice 1) y un último píxel adyacente al segundo borde del bloque de datos de video (por ej., el píxel más a la derecha de la fila 268, que tiene un valor de índice 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 en la segunda línea. Esta situación (es decir, donde el valor de índice del último píxel en la primera línea es el mismo que el 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 computadora de otros tipos de contenido de video.

[00181] En algunos ejemplos, el codificador de video 20 puede utilizar un orden

de exploración de serpiente cuando se codifican los índices del mapa. Por ejemplo, el codificador de video 20 puede escanear el último píxel de la segunda línea directamente después del último píxel de la primera línea. De esta manera, el codificador de video 20 puede mejorar la eficiencia de la codificación de longitud de ejecución.

[00182] Por ejemplo, en lugar de utilizar un orden de exploración de trama, el codificador de video 20 puede utilizar un orden de exploración de serpiente para codificar los valores del mapa 240. En un ejemplo con fines de ilustración, si se consideran las filas 266, 268, y 270 del mapa 240, por el uso de un orden de exploración de serpiente (tal como un orden de exploración de serpiente), el codificador de video 20 puede codificar los valores del mapa 240 que comienzan con la posición izquierda de la fila 266, procediendo a través de la posición más a la derecha de la fila 266, luego hacia abajo hasta la posición más a la izquierda de la fila 268, procediendo a través de la posición más a la izquierda de la fila 268, y luego hacia el extremo izquierdo de la fila 270. Por ejemplo, el codificador de video 20 puede codificar uno o más elementos de sintaxis que indican que la primera posición de la fila 266 es uno y que la siguiente ejecución de dos entradas consecutivas en la dirección de exploración son las mismas que la primera posición de la fila 266.

[00183] El codificador de video 20 puede codificar uno o más elementos de sintaxis 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 video 20 puede codificar uno o más elementos de sintaxis que indican que la siguiente posición de la fila 266 (es decir, la sexta posición) es tres, y que la siguiente ejecución de cinco entradas consecutivas en la dirección de exploración es la misma que la sexta posición de la fila 266. El codificador de video 20 puede codificar uno o más elementos de sintaxis 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 la próxima ejecución de nueve entradas consecutivas en la dirección de exploración es la misma que la cuarta posición de la fila 268.

[00184] De esta manera, por el uso de un orden de exploración de serpiente, el codificador de video 20 puede codificar ejecuciones de mayor longitud, que pueden mejorar la eficiencia de codificación. Por ejemplo, por el uso de la exploración de trama, la ejecución final de la fila 266 (para el valor de índice 3) es igual a 2. Por el uso de la exploración de serpiente, sin embargo, la ejecución final de la fila 266 se

extiende en la fila 268 y es igual a 5.

[00185] El decodificador de video 30 puede recibir los elementos de sintaxis descriptos con anterioridad y reconstruir las filas 266, 268 y 270. Por ejemplo, el decodificador de video 30 puede obtener, a partir de una corriente de bits codificada, los datos que indican un valor de índice para una posición del mapa 240 que actualmente está siendo codificado. El decodificador de video 30 también puede obtener datos que indican el número de posiciones consecutivas en el orden de exploración que tienen el mismo valor de índice.

[00186] La FIG. 6 es un diagrama de flujo que ilustra un proceso representativo para la decodificación de un bloque de datos de video por el uso del modo de paleta, de acuerdo con una o más técnicas de esta revelación. Las técnicas de la FIG. 6 pueden ser llevadas a cabo por un decodificador de video, tales como el decodificador de video 30 que se ilustra en la FIG. 1 y la FIG. 3. Para fines de ilustración, las técnicas de la FIG. 6 se describen en el contexto del decodificador de video 30 de la FIG. 1 y la FIG. 3, si bien los decodificadores de video que tienen configuraciones diferentes que la de decodificador de video 30 pueden llevar a cabo las técnicas de la FIG. 6.

[00187] De acuerdo con lo discutido con anterioridad, puede ser deseable maximizar el número de compartimientos codificados por el modo de derivación de elementos de sintaxis que se agrupan juntos. De acuerdo con una o más técnicas de esta revelación, el decodificador de video 30 puede decodificar, a partir de una corriente de bits de video codificado y por el uso del modo de derivación, un grupo de elementos de sintaxis para una paleta para un bloque actual de datos de video (602). Por ejemplo, la unidad de decodificación de la entropía 150 del decodificador de video 30 puede decodificar, por el uso del modo de derivación, compartimientos de uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual (por ej., uno o más elementos de sintaxis **palette_predictor_run**), un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas o inferidas en forma explícita (por ej., un elemento de sintaxis **num_signalled_palette_entries**), uno o más elementos de sintaxis que indican cada uno un valor de un componente en una entrada en la paleta actual (por ej., uno o más elementos de sintaxis **palette_entry**), un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada (por ej., un elemento de sintaxis

palette_escape_val_present_flag), un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas o inferidas en forma explícita (por ej., un elemento de sintaxis **num_palette_indices_idc**), y uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales (por ej., uno o más elementos de sintaxis **palette_index_idc**). En algunos ejemplos, para decodificar un grupo de elementos de sintaxis codificados por derivación, el decodificador de video 30 puede decodificar secuencialmente los elementos de sintaxis incluidos en el grupo de elementos de sintaxis sin decodificar ningún compartimiento codificado sin derivación. De acuerdo con lo discutido con anterioridad, el agrupamiento de un gran número de compartimientos/elementos de sintaxis codificados por derivación puede mejorar un rendimiento de CABAC del decodificador de video 30. En particular, el agrupamiento de elementos de sintaxis codificados por derivación puede permitir que el decodificador de video 30 evite iniciar/detener/reiniciar el motor CABAC. Por el contrario, cuando los elementos de sintaxis codificados por derivación no se agrupan, el decodificador de video 30 puede tener que iniciar continuamente el motor CABAC para decodificar un compartimiento codificado sin derivación con un primer contexto, detener el motor CABAC para decodificar un compartimiento codificado por derivación, iniciar el motor CABAC para decodificar otro compartimiento codificado sin derivación con el primer contexto, etc. De acuerdo con lo discutido con anterioridad, la conmutación repetida del motor CABAC puede disminuir el rendimiento del motor de CABAC.

[00188] El decodificador de video 30 puede decodificar, por el uso de CABAC con un contexto y en una posición en la corriente de bits de video codificado que se encuentra después del grupo de elementos de sintaxis, un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video (604). Por ejemplo, la unidad de decodificación de la entropía 150 del decodificador de video 30 puede decodificar, por el uso de CABAC con un contexto, el compartimiento de un elemento de sintaxis **palette_transpose_flag**.

[00189] El decodificador de video 30 puede decodificar, por el uso de CABAC con un contexto y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video, uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video (606).

Por ejemplo, la unidad de decodificación de la entropía 150 del decodificador de video 30 puede decodificar, por el uso de CABAC con uno o más contextos, compartimientos de un elemento de sintaxis que especifica el valor absoluto de la diferencia entre un QP (por ej., un QP luma) para el bloque actual de datos de video y un predictor del QP para el bloque actual (por ej., **cu_qp_delta_abs**), un elemento de sintaxis que especifica un signo de la diferencia entre el QP para el bloque actual de datos de video y el predictor del QP para el bloque actual (por ej., **cu_qp_delta_sign_flag**), un elemento de sintaxis que indica si las entradas en una o más listas de compensaciones se añaden a un QP luma para el bloque actual para determinar los QP croma para el bloque actual (por ej., **cu_chroma_qp_offset_flag**), y un elemento de sintaxis que especifica un índice de una entrada en cada una de las una o más listas de compensaciones que se agregan al QP luma para el bloque actual para determinar los QP croma para el bloque actual (por ej., **cu_chroma_qp_offset_idx**).

[00190] En algunos ejemplos, el decodificador de video 30 puede decodificar los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video con base en un valor de un elemento de sintaxis del grupo de elementos de sintaxis decodificado por el uso del modo de derivación. Como un ejemplo, el decodificador de video 30 puede decodificar los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video en el que el elemento de sintaxis del grupo de elementos de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada que indica que el bloque actual de datos de video no incluye por lo menos una muestra de escape. Como otro ejemplo, el decodificador de video 30 puede no decodificar los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video en el que el elemento de sintaxis del grupo de elementos de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada indica que el bloque actual de datos de video no incluye por lo menos una muestra de escape.

[00191] El decodificador de video 30 puede generar la paleta para el bloque actual de datos de video con base en el grupo de elementos de sintaxis y el elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video (608) y decodificar el bloque actual de datos de video basado en paletas generada y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de

datos de video (610). Por ejemplo, la unidad de decodificación basada en paletas 165 puede generar la paleta que tiene entradas que indican los valores de píxel, recibir información que asocia por lo menos algunas posiciones del bloque actual de datos de video con entradas en la paleta, seleccionar valores de píxel de la paleta con base en la información, y reconstruir los valores de píxel del bloque con base en los valores de píxel seleccionados.

[00192] La FIG. 7 es un diagrama de flujo que ilustra un proceso representativo para la codificación de un bloque de datos de video por el uso del modo de paleta, de acuerdo con una o más técnicas de esta revelación. Las técnicas de la FIG. 7 pueden ser llevadas a cabo por un codificador de video, tal como el codificador de video 20 que se ilustra en la FIG. 1 y la FIG. 2. Para fines de ilustración, las técnicas de la FIG. 7 se describen en el contexto de codificador de video 20 de la FIG. 1 y la FIG. 2, si bien los codificadores de video que tienen configuraciones diferentes que la del codificador de video 20 puede llevar a cabo las técnicas de la FIG. 7.

[00193] De acuerdo con lo discutido con anterioridad, puede ser deseable maximizar el número de compartimientos codificados por el modo de derivación de elementos de sintaxis que se agrupan juntos. De acuerdo con una o más técnicas de esta revelación, el codificador de video 20 puede codificar, en una corriente de bits de video codificado y por el uso de modo de derivación, un grupo de elementos de sintaxis para una paleta para un bloque actual de datos de video (702). Por ejemplo, la unidad de codificación de la entropía 118 del codificador de video 20 puede codificar, por el uso del modo de derivación, compartimientos de uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual (por ej., uno o más elementos de sintaxis **palette_predictor_run**), un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas o inferidas en forma explícita (por ej., un elemento de sintaxis **num_signalled_palette_entries**), uno o más elementos de sintaxis que indican cada uno un valor de un componente en una entrada en la paleta actual (por ej., uno o más elementos de sintaxis **palette_entry**), un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada (por ej., un elemento de sintaxis **palette_escape_val_present_flag**), un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas o inferidas en forma explícita (por ej., un elemento de sintaxis **num_palette_indices_idc** o

num_palette_indices_minus1), y uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales (por ej., uno o más elementos de sintaxis **palette_index_idc**).

[00194] El codificador de video 20 puede codificar, por el uso de CABAC con un contexto y en una posición en la corriente de bits de video codificado que se encuentra después del grupo de elementos de sintaxis, un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video (704). Por ejemplo, la unidad de codificación de la entropía 118 del codificador de video 20 puede codificar, por el uso de CABAC con un contexto, el compartimiento de un elemento de sintaxis **palette_transpose_flag**.

[00195] El codificador de video 20 puede codificar, por el uso de CABAC con un contexto y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video, uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video (706). Por ejemplo, la unidad de codificación de la entropía 118 del codificador de video 20 puede codificar, por el uso de CABAC con uno o más contextos, los compartimientos de un elemento de sintaxis que especifica el valor absoluto de la diferencia entre un QP luma para el bloque actual de datos de video y un predictor del QP luma para el bloque actual (por ej., **cu_qp_delta_abs**), un elemento de sintaxis que especifica un signo de la diferencia entre el QP luma para el bloque actual de datos de video y el predictor del QP luma para el bloque actual (por ej., **cu_qp_delta_sign_flag**), un elemento de sintaxis que indica si las entradas en una o más listas de compensaciones se añaden al QP luma para el bloque actual para determinar los QP croma para el bloque actual (por ej., **cu_chroma_qp_offset_flag**), y un elemento de sintaxis que especifica un índice de una entrada en cada una de las una o más listas de compensaciones que se agregan al QP luma para el bloque actual para determinar los QP croma para el bloque actual (por ej., **cu_chroma_qp_offset_idx**).

[00196] En algunos ejemplos, el codificador de video 20 puede codificar los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video con base en un valor de un elemento de sintaxis del grupo de elementos de sintaxis codificados en el modo de derivación. Como un ejemplo, el codificador de video 20 puede codificar los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el

bloque actual de datos de video en el que el elemento de sintaxis del grupo de elementos de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada indica que el bloque actual de datos de video no incluye por lo menos una muestra de escape. Como otro ejemplo, el codificador de video 20 puede no codificar los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video en el que el elemento de sintaxis del grupo de elementos de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada indica que el bloque actual de datos de video no incluye por lo menos una muestra de escape.

[00197] Se ha de reconocer que, dependiendo del ejemplo, se pueden añadir ciertos actos o eventos de cualquiera de las técnicas descriptas en la presente memoria se pueden llevar a cabo en una secuencia diferente, se pueden combinar, o se pueden excluir por completo (por ej., no todos los actos o eventos descriptos son necesarios para la práctica de las técnicas). Además, en algunos ejemplos, los actos o eventos pueden ser llevadas a cabo en forma simultánea, por ejemplo, a través de un procesamiento de múltiples hilos, un procesamiento de interrupción, o múltiples procesadores, en lugar de secuencialmente. Además, si bien ciertos aspectos de esta revelación se describen como realizados por un único módulo o unidad para fines de claridad, se debe entender que las técnicas de esta revelación se pueden llevar a cabo por medio de una combinación de unidades o módulos asociados con un codificador de video.

[00198] Ciertos aspectos de esta revelación se han descripto con respecto al estándar de HEVC en desarrollo para fines de ilustración. Sin embargo, las técnicas descriptas en esta revelación pueden ser útiles para otros procesos de codificación de video, que incluyen otros procesos de codificación de video estándares o de propiedad aún no desarrollados.

[00199] Las técnicas descriptas con anterioridad pueden ser llevadas a cabo por el codificador de video 20 (FIGS. 1 y 2) y/o el decodificador de video 30 (FIGS. 1 y 3), ambos de los cuales se pueden denominar por lo general como un codificador de video. Del mismo modo, la codificación de video se puede referir a la codificación de video o la decodificación de video, de acuerdo con lo aplicable.

[00200] Si bien las combinaciones particulares de diversos aspectos de las técnicas se han descripto con anterioridad, estas combinaciones se proporcionan simplemente para ilustrar ejemplos de las técnicas descriptas en esta revelación. En

consecuencia, las técnicas de esta revelación no se deben limitar a estos ejemplos de combinaciones y pueden abarcar cualquier combinación concebible de los diversos aspectos de las técnicas descriptas en esta revelación.

[00201] En uno o más ejemplos, las funciones descriptas se pueden implementar en hardware, software, firmware, o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en o transmitir a través de, como una o más instrucciones o código, un medio legible por computadora y ejecutadas por una unidad de procesamiento basada en hardware. Los medios legibles por computadora pueden incluir medios de almacenamiento legibles por computadora, lo que corresponde a un medio tangible tal como un medio de almacenamiento de datos, o un medio de comunicación que incluye cualquier medio que facilite la transferencia de un programa de computadora de un lugar a otro, por ej., de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por computadora por lo general pueden corresponder a (1) medios de almacenamiento legibles por computadora tangibles que no son transitorios o (2) un medio de comunicación tal como una señal u onda de portador. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se puede acceder por medio de una o más computadoras o uno o más procesadores para recuperar instrucciones, códigos y/o estructuras de datos para la aplicación de las técnicas descriptas en esta revelación. Un producto de programa de computadora puede incluir un medio legible por computadora.

[00202] A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por computadora 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 puede utilizar para almacenar el código del programa deseado en forma de instrucciones o estructuras de datos y que se puede acceder por medio de una computadora. Además, cualquier conexión se denomina correctamente un medio legible por computadora. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor, u otra fuente remota por el uso de un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidas en la definición de medio. Se debe entender, sin embargo, que los medios de almacenamiento legibles por computadora y los medios de

almacenamiento de datos no incluyen conexiones, ondas portadoras, señales, u otros medios de comunicación transitorios, sino que se dirigen a medios de almacenamiento no transitorios, tangibles. Los términos disco 1 y disco 2, de acuerdo con lo utilizado en la presente memoria, incluye disco compacto (CD), discos láser, discos ópticos, discos versátiles digitales (DVD), disquetes y discos Blu-ray, donde los discos 1 por lo general reproducen datos en forma magnética, mientras que los discos 2 reproducen datos en forma óptica con láseres. Las combinaciones de los anteriores también se deberían incluir dentro del alcance de medios legibles por computadora.

[00203] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores digitales de señal (DSP), microprocesadores de propósito general, Circuitos Integrados de Aplicación Específica (ASIC), Matrices de Puertas Programables de Campo (FPGA), u otros circuitos equivalentes integrados o lógica discreta. De acuerdo con ello, el término "procesador", de acuerdo con lo utilizado en la presente memoria se puede referir a cualquiera de la estructura anterior o cualquier otra estructura adecuada para la aplicación de las técnicas descriptas en la presente memoria. Además, en algunos aspectos, la funcionalidad descripta en la presente memoria se puede proporcionar dentro de los módulos de hardware y/o software dedicado configurados para codificar y decodificar, o incorporado en un códec combinado. Además, las técnicas se podrían aplicar plenamente en uno o más circuitos o elementos lógicos.

[00204] Las técnicas de esta revelación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de circuitos integrados (por ej., un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta revelación para enfatizar los aspectos funcionales de los dispositivos configurados para llevar a cabo las técnicas reveladas, pero no necesariamente requieren la realización de diferentes unidades de hardware. Más bien, de acuerdo con lo descripto con anterioridad, las diversas unidades pueden ser combinadas en una unidad de hardware códec o proporcionados por una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores de acuerdo con lo descripto con anterioridad, en conjunción con el software y/o firmware adecuado.

[00205] Se han descripto varios ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un método de decodificación de datos de video, el método comprende:

la decodificación, a partir de una corriente de bits de video codificado y el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto, de un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video;

la decodificación, a partir de la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video y el uso de CABAC con un contexto, de uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video;

la decodificación, a partir de la corriente de bits de video codificado, un grupo de elementos de sintaxis por el uso del modo de derivación, en el que el grupo comprende:

uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual;

un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita;

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

un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada;

un elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita; y

uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales; y

la decodificación del bloque actual de datos de video basado en paletas para el bloque actual de datos de video, el grupo de elementos de sintaxis y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

2. El método de acuerdo con la reivindicación 1, en el que el elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque

actual de datos de video comprende un elemento de sintaxis palette_transpose_flag.

3. El método de acuerdo con la reivindicación 1, en el que los uno o más elementos de sintaxis relacionados con QP delta comprenden uno o ambos de un elemento de sintaxis 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 de sintaxis que indica un signo de la diferencia entre el QP del bloque actual y el predictor del QP del bloque actual.

4. El método de acuerdo con la reivindicación 1, en el que los uno o más elementos de sintaxis relacionados con compensaciones de QP croma comprenden uno o ambos de un elemento de sintaxis que indica si las entradas en una o más listas de compensaciones se añaden a un QP luma del bloque actual para determinar los QP croma para el bloque actual y un elemento de sintaxis que indica un índice de una entrada en cada una de las una o más listas de compensaciones que se agregan al QP luma para el bloque actual para determinar los QP croma para el bloque actual.

5. El método de acuerdo con la reivindicación 1, en el que uno o más de:

los uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual comprenden uno o más elementos de sintaxis palette_predictor_run,

el elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita comprende un elemento de sintaxis num_signalled_palette_entries,

los uno o más elementos de sintaxis que indican cada uno un valor de un componente en una entrada en la paleta actual comprenden uno o más elementos de sintaxis palette_entry,

el elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos un muestra de escape codificada comprende palette_escape_val_present_flag,

el elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita comprende un elemento de sintaxis num_palette_indices_idc, y

los uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales comprenden uno o más elementos de sintaxis palette_index_idc.

6. El método de acuerdo con la reivindicación 1, en el que la decodificación del

grupo de elementos de sintaxis comprende la decodificación del grupo de elementos de sintaxis de la corriente de bits de video codificado en una posición en la corriente de bits de video codificado que se encuentra antes del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video.

7. El método de acuerdo con la reivindicación 1, que además comprende:

la decodificación, a partir de la corriente de bits de video codificado después de que el grupo de elementos de sintaxis se codifica por el uso del modo de derivación, un elemento de sintaxis que indica una última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video.

8. El método de acuerdo con la reivindicación 7, en el que la decodificación del elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video comprende la decodificación del elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video por el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto.

9. Un método de codificación de datos de video, el método comprende:

la codificación, en una corriente de bits de video codificado y el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto, de un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video;

la codificación, en la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video y el uso de CABAC con un contexto, de uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video;

la codificación, a partir de la corriente de bits de video codificado, un grupo de elementos de sintaxis por el uso del modo de derivación, en el que el grupo comprende:

uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual;

un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita;

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

un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada;

un elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita; y

uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales; y

la codificación del bloque actual de datos de video basado en paletas para el bloque actual de datos de video, el grupo de elementos de sintaxis y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

10. El método de acuerdo con la reivindicación 9, en el que el elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video comprende un elemento de sintaxis palette_transpose_flag.

11. El método de acuerdo con la reivindicación 9, en el que los uno o más elementos de sintaxis relacionados con QP delta comprenden uno o ambos de un elemento de sintaxis 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 de sintaxis que indica un signo de la diferencia entre el QP del bloque actual y el predictor del QP del bloque actual.

12. El método de acuerdo con la reivindicación 9, en el que los uno o más elementos de sintaxis relacionados con compensaciones de QP croma comprenden uno o ambos de un elemento de sintaxis que indica si las entradas en una o más listas de compensaciones se añaden a un QP luma del bloque actual para determinar los QP croma para el bloque actual y un elemento de sintaxis que indica un índice de una entrada en cada una de las una o más listas de compensaciones que se agregan al QP luma para el bloque actual para determinar los QP croma para el bloque actual.

13. El método de acuerdo con la reivindicación 9, en el que uno o más de:

los uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual comprenden uno o más elementos de sintaxis, palette_predictor_run,

el elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita comprende un elemento de sintaxis

num_signalled_palette_entries,

los uno o más elementos de sintaxis que indican cada uno un valor de un componente en una entrada en la paleta actual comprenden uno o más elementos de sintaxis palette_entry,

el elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada comprende palette_escape_val_present_flag,

el elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita comprenden un elemento de sintaxis num_palette_indices_minus1, y

los uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales comprenden uno o más elementos de sintaxis palette_index_idc.

14. El método de acuerdo con la reivindicación 9, en el que la codificación del grupo de elementos de sintaxis comprende la codificación del grupo de elementos de sintaxis en la corriente de bits de video codificado en una posición en la corriente de bits de video codificado que se encuentra antes del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video.

15. El método de acuerdo con la reivindicación 9, que además comprende:

la codificación, en la corriente de bits de video codificado después de que el grupo de elementos de sintaxis se codifica por el uso del modo de derivación, de un elemento de sintaxis que indica una última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video.

16. El método de acuerdo con la reivindicación 15, en el que la codificación del elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video comprende la codificación del elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video por el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto.

17. Un dispositivo para la codificación o decodificación de datos de video, el dispositivo comprende:

una memoria configurada para almacenar datos de video;

uno o más procesadores configurados para:

codificar o decodificar, en una corriente de bits de video codificado y el uso de

codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto, un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video;

codificar o decodificar, en la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video y el uso de CABAC con un contexto, uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video;

codificar o decodificar, a partir de la corriente de bits de video codificado, un grupo de elementos de sintaxis por el uso del modo de derivación, en el que el grupo comprende:

uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual;

un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita;

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

un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada;

un elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita; y

uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales; y

codificar o decodificar el bloque actual de datos de video basado en paleta para el bloque actual de datos de video, el grupo de elementos de sintaxis y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.

18. El dispositivo de acuerdo con la reivindicación 17, en el que el elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video comprende un elemento de sintaxis palette_transpose_flag.

19. El dispositivo de acuerdo con la reivindicación 17, en el que uno o más de:

los uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual comprenden uno o más elementos de sintaxis, palette_predictor_run,

el elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita comprende un elemento de sintaxis num_signalled_palette_entries,

los uno o más elementos de sintaxis que indican cada uno un valor de un componente en una entrada en la paleta actual comprenden uno o más elementos de sintaxis palette_entry,

el elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada que comprende palette_escape_val_present_flag,

el elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas o inferidas en forma explícita comprende un elemento de sintaxis num_palette_indices_minus1, y

los uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales comprenden uno o más elementos de sintaxis palette_index_idc.

20. El dispositivo de acuerdo con la reivindicación 17, en el que, para codificar o decodificar el grupo de elementos de sintaxis, los uno o más procesadores están configurados para codificar o decodificar el grupo de elementos de sintaxis en la corriente de bits de video codificado en una posición en la corriente de bits de video codificado que se encuentra antes del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta del bloque actual de datos de video.

21. El dispositivo de acuerdo con la reivindicación 17, en el que los uno o más procesadores están configurados además para:

codificar o decodificar, en la corriente de bits de video codificado después de que el grupo de elementos de sintaxis se codifica por el uso del modo de derivación, un elemento de sintaxis que indica una última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video.

22. El dispositivo de acuerdo con la reivindicación 21, en el que, para codificar o decodificar el elemento de sintaxis que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video, los uno o más procesadores están configurados para codificar o decodificar el elemento de sintaxis

que indica la última aparición de un marcador de tipo de ejecución dentro del bloque actual de datos de video por el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto.

23. Un dispositivo para la decodificación de datos de video, el dispositivo comprende:

un medio para la decodificación, a partir de una corriente de bits de video codificado y el uso de codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto, de un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video;

un medio para la decodificación, a partir de la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video y el uso de CABAC con el contexto, de uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video;

un medio para la decodificación, a partir de la corriente de bits de video codificado, un grupo de elementos de sintaxis por el uso del modo de derivación, en el que el grupo comprende:

uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual;

un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita;

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

un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada;

un elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita; y

uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales; y

un medio para la decodificación del bloque actual de datos de video basado en paletas para el bloque actual de datos de video, el grupo de elementos de sintaxis y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta

y/o QP croma para el bloque actual de datos de video.

24. Un medio de almacenamiento legible por computadora que almacena por lo menos una porción de una corriente de bits de video codificado que, cuando se procesa por medio de un dispositivo de decodificación de video, provoca que uno o más procesadores del dispositivo de decodificación de video:

determinen si un proceso de transposición se aplica a los índices de paleta de una paleta para un bloque actual de datos de video;

generen la paleta para el bloque actual de los datos de video basados en:

uno o más elementos de sintaxis que indican un número de ceros que preceden a una entrada distinta de cero en una matriz que indica si las entradas de una paleta de predicción se vuelven a utilizar en la paleta actual;

un elemento de sintaxis que indica un número de entradas en la paleta actual que están señalizadas en forma explícita;

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

un elemento de sintaxis que indica si el bloque actual de datos de video incluye por lo menos una muestra de escape codificada;

un elemento de sintaxis que indica una serie de índices en la paleta actual que están señalizados o inferidos en forma explícita; y

uno o más elementos de sintaxis que indican los índices en una matriz de entradas de paleta actuales; y

decodifiquen el bloque actual de los datos de video basado en paletas para el bloque actual de datos de video y un parámetro de cuantificación delta (QP) y una o más compensaciones de QP croma para el bloque actual de datos de video,

en el que uno o más elementos de sintaxis relacionados con el QP delta y uno o más elementos de sintaxis relacionados con las una o más compensaciones de QP croma para el bloque actual de datos de video se encuentran en una posición en la corriente de bits de video codificado que se encuentra después de un elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video, en el que el elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video se decodifica usando la codificación aritmética binaria adaptativa al contexto (CABAC) con un contexto, en el que al menos uno de los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y uno o más elementos de sintaxis relacionados con compensaciones de QP croma se

decodifican utilizando CABAC con un contexto, y en el que el grupo de elementos de sintaxis se decodifican utilizando un modo de derivación.

RESUMEN

Un método representativo de codificación de datos de video incluye la codificación, a partir de una corriente de bits de video codificado, un elemento de sintaxis que indica si un proceso de transposición se aplica a índices de paleta de una paleta para un bloque actual de datos de video; la decodificación, de la corriente de bits de video codificado y en una posición en la corriente de bits de video codificado que se encuentra después del elemento de sintaxis que indica si el proceso de transposición se aplica a índices de paleta de la paleta para el bloque actual de datos de video, uno o más elementos de sintaxis relacionados con compensaciones de un parámetro de cuantificación delta (QP) y/o QP croma para el bloque actual de datos de video; y la decodificación del bloque actual de datos de video basado en paletas para el bloque actual de datos de video y los uno o más elementos de sintaxis relacionados con compensaciones de QP delta y/o QP croma para el bloque actual de datos de video.