

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 841 312**

51 Int. Cl.:

H04N 19/52 (2014.01)

H04N 19/537 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **05.10.2017 PCT/US2017/055350**

87 Fecha y número de publicación internacional: **12.04.2018 WO18067823**

96 Fecha de presentación y número de la solicitud europea: **05.10.2017 E 17784818 (1)**

97 Fecha y número de publicación de la concesión europea: **30.09.2020 EP 3523972**

54 Título: **Predicción de vectores de movimiento para modelos de movimiento afines en codificación de vídeo**

30 Prioridad:

05.10.2016 US 201662404719 P

04.10.2017 US 201715725052

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

08.07.2021

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)

5775 Morehouse Drive

San Diego, CA 92121-1714, US

72 Inventor/es:

CHEN, YI-WEN;

CHIEN, WEI-JUNG;

ZHANG, LI;

SUN, YU-CHEN;

CHEN, JIANLE y

KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 841 312 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Predicción de vectores de movimiento para modelos de movimiento afines en codificación de vídeo

5 CAMPO TÉCNICO

[0001] Esta divulgación se refiere a dispositivos, sistemas y procedimientos para la codificación de vídeo.

ANTECEDENTES

10 [0002] Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, incluyendo televisiones digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas electrónicas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, codificación avanzada de vídeo (AVC), el ITU-T-H.265, el estándar de codificación de vídeo de alta eficacia (HEVC) y las extensiones de dichos estándares. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente implementando dichas técnicas de compresión de vídeo.

25 [0003] Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. En la codificación de vídeo basada en bloques, un fragmento de vídeo (es decir, una trama de vídeo o una parte de una trama de vídeo) se puede dividir en bloques de vídeo, que también se pueden denominar bloques en árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia de bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales se pueden transformar desde el dominio de píxel a un dominio de transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden cuantificar. La técnica anterior muestra un documento creado por HUAWEI TECHNOLOGIES: "Affine transform prediction for next generation video", REUNIÓN UIT-T SG16; 12-10-2015-23-10-2015; GINEBRA, n.º T13-SG16-C-1016, 29 de septiembre de 2015 (29-09-2015), documento XP030100743. Este documento explica que en el mundo real hay muchos tipos de movimientos, por ejemplo, acercar/alejar, rotación, movimientos de perspectiva y otros movimientos irregulares, y presenta una herramienta de predicción de transformada afín simplificada.

45 BREVE EXPLICACIÓN

[0004] La invención se define como se expone en las reivindicaciones adjuntas. En general, esta divulgación describe técnicas relacionadas con la predicción del vector de movimiento y la reconstrucción del vector de movimiento para el modo de predicción de movimiento afín. Las técnicas se pueden aplicar a cualquiera de los códecs de vídeo existentes, tales como HEVC (codificación de vídeo de alta eficacia) o pueden ser una herramienta de codificación eficaz en cualquier estándar de codificación de vídeo futuro.

55 [0005] En un ejemplo, esta divulgación describe un procedimiento de descodificación de datos de vídeo, comprendiendo el procedimiento: seleccionar un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es espacialmente vecino a un bloque actual; extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual; insertar, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afines que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual; determinar, en base a un índice señalado en un flujo de bits, un conjunto de MVP afín seleccionado en la lista de candidatos de conjuntos de MVP afines; obtener, a partir del flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y predictores de vectores de movimiento en el conjunto de MVP afín seleccionado; y determinar, en base a los predictores de vectores de movimiento incluidos en el conjunto de MVP afín seleccionado y las MVD, vectores de movimiento de los puntos de control del bloque actual; generar, en base a los vectores de movimiento de los puntos de control del bloque actual, un bloque predictivo; y reconstruir el bloque actual en base a datos residuales y el bloque predictivo.

[0006] En otro ejemplo, esta divulgación describe un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento: seleccionar un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es espacialmente vecino a un bloque actual; extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual; insertar, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afines que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual; seleccionar un conjunto de MVP afín en la lista de candidatos de conjuntos de MVP afines; señalar, en un flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y predictores de vectores de movimiento en el conjunto de MVP afín seleccionado; y señalar, en el flujo de bits, un índice que indica una posición en la lista de candidatos de conjuntos de MVP afines del conjunto de MVP afín seleccionado.

[0007] En otro ejemplo, esta divulgación describe un dispositivo para decodificar datos de vídeo, comprendiendo el dispositivo: una memoria configurada para almacenar los datos de vídeo; y uno o más circuitos de procesamiento configurados para: seleccionar un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es espacialmente vecino a un bloque actual; extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual; insertar, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afines que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual; determinar, en base a un índice señalado en un flujo de bits, un conjunto de MVP afín seleccionado en la lista de candidatos de conjuntos de MVP afines; obtener, a partir del flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y predictores de vectores de movimiento en el conjunto de MVP afín seleccionado; y determinar, en base a los predictores de vectores de movimiento incluidos en el conjunto de MVP afín seleccionado y las MVD, vectores de movimiento de los puntos de control del bloque actual; generar, en base a los vectores de movimiento de los puntos de control del bloque actual, un bloque predictivo; y reconstruir el bloque actual en base a datos residuales y el bloque predictivo.

[0008] En otro ejemplo, esta divulgación describe un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo: una memoria configurada para almacenar los datos de vídeo; y uno o más circuitos de procesamiento configurados para: seleccionar un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es espacialmente vecino a un bloque actual; extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual; insertar, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afines que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual; seleccionar un conjunto de MVP afín en la lista de candidatos de conjuntos de MVP afines; señalar, en un flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y predictores de vectores de movimiento en el conjunto de MVP afín seleccionado; y señalar, en el flujo de bits, un índice que indica una posición en la lista de candidatos de conjuntos de MVP afines del conjunto de MVP afín seleccionado.

[0009] En otro ejemplo, esta divulgación describe un dispositivo para decodificar datos de vídeo, comprendiendo el dispositivo: medios para seleccionar un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es espacialmente vecino a un bloque actual; medios para extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual; medios para insertar, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afines que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual; medios para determinar, en base a un índice señalado en un flujo de bits, un conjunto de MVP afín seleccionado en la lista de candidatos de conjuntos de MVP afines; medios para obtener, a partir del flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y predictores de vectores de movimiento en el conjunto de MVP afín seleccionado; medios para determinar, en base a los predictores de vectores de movimiento incluidos en el conjunto de MVP afín seleccionado y las MVD, vectores de movimiento de los puntos de control del bloque actual; medios para generar, en base a los vectores de movimiento de los puntos de control del bloque actual, un bloque predictivo; y medios para reconstruir el bloque actual en base a datos residuales y el bloque predictivo.

[0010] En otro ejemplo, esta divulgación describe un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo: medios para seleccionar un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es espacialmente vecino a un bloque actual; medios para extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual; medios para insertar, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afines que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual; medios para seleccionar un conjunto de MVP afín en la lista de candidatos de conjuntos de MVP afines; medios para señalar, en un flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y predictores de vectores de movimiento en el conjunto de MVP afín seleccionado; y medios para señalar, en el flujo de bits, un índice que indica una posición en la lista de candidatos de conjuntos de MVP afines del conjunto de MVP afín seleccionado.

[0011] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más circuitos de procesamiento de un dispositivo para decodificar vídeo: seleccionen un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es espacialmente vecino a un bloque actual; extrapolen vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual; inserten, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afines que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual; determinen, en base a un índice señalado en un flujo de bits, un conjunto de MVP afín seleccionado en la lista de candidatos de conjuntos de MVP afines; obtengan, a partir del flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y predictores de vectores de movimiento en el conjunto de MVP afín seleccionado; determinen, en base a los predictores de vectores de movimiento incluidos en el conjunto de MVP afín seleccionado y las MVD, vectores de movimiento de los puntos de control del bloque actual; generen, en base a los vectores de movimiento de los puntos de control del bloque actual, un bloque predictivo; y reconstruyan el bloque actual en base a datos residuales y el bloque predictivo.

[0012] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más circuitos de procesamiento de un dispositivo para codificar datos de vídeo: seleccionen un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es espacialmente vecino a un bloque actual; extrapolen vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual; inserten, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afines que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual; seleccionen un conjunto de MVP afín en la lista de candidatos de conjuntos de MVP afines; señalicen, en un flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y predictores de vectores de movimiento en el conjunto de MVP afín seleccionado; y señalicen, en el flujo de bits, un índice que indica una posición en la lista de candidatos de conjuntos de MVP afines del conjunto de MVP afín seleccionado.

[0013] Los detalles de uno o más aspectos de la divulgación se exponen en las figuras adjuntas y en la descripción a continuación. Otros rasgos característicos, objetivos y ventajas de las técnicas descritas en esta divulgación resultarán evidentes a partir de la descripción, las figuras y las reivindicaciones.

BREVE DESCRIPCIÓN DE LAS FIGURAS

[0014]

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo ejemplar que puede utilizar una o más técnicas descritas en esta divulgación.

La FIG. 2A ilustra candidatos de vector de movimiento (MV) vecino espacial para el modo de fusión.

La FIG. 2B ilustra candidatos de MV vecinos espaciales para el modo de predicción de vector de movimiento avanzado (AMVP).

La FIG. 3A es un diagrama conceptual que ilustra una técnica ejemplar para derivar un candidato de predictor de vector de movimiento temporal (TMVP).

La FIG. 3B ilustra un ajuste a escala de vector de movimiento.

La FIG. 4 ilustra un modelo de movimiento afín simplificado para un bloque actual.

La FIG. 5 ilustra un campo de vector de movimiento afín (MVF) por subbloque.

La FIG. 6A es un diagrama de bloques que ilustra un bloque actual y bloques vecinos como se usa en el modo AF_INTER.

La FIG. 6B ilustra un ejemplo de lista de candidatos de conjunto de MVP afín usada en un modelo de movimiento afín de 4 parámetros.

La FIG. 7A muestra bloques vecinos usados al codificar un bloque actual en el modo AF_MERGE.

La FIG. 7B ilustra candidatos para AF_MERGE.

La FIG. 8A ilustra bloques ejemplares usados en un modelo de movimiento afín de 6 parámetros.

La FIG. 8B ilustra un ejemplo de lista de candidatos de conjunto de MVP afín usada en un modelo de movimiento afín de 6 parámetros.

5 La FIG. 9 ilustra una lista de candidatos de conjunto de MVP afín ejemplar que incluye un conjunto de MVP extrapolado, de acuerdo con una técnica de esta divulgación.

10 La FIG. 10 ilustra predicción de movimiento de subbloque o predicción de parámetro, de acuerdo con una técnica de esta divulgación, donde el movimiento afín de cada subbloque de un bloque actual se puede predecir o heredar directamente del movimiento extrapolado de sus propios bloques vecinos.

La FIG. 11A ilustra una lista de candidatos de conjunto de MVP afín ejemplar para un modelo de movimiento afín de 4 parámetros, de acuerdo con una técnica de esta divulgación.

15 La FIG. 11B ilustra una lista de candidatos de conjunto de MVP afín ejemplar para un modelo de movimiento afín de 6 parámetros, de acuerdo con una técnica de esta divulgación.

La FIG. 12 es un diagrama de bloques que ilustra un codificador de vídeo ejemplar que puede implementar una o más técnicas descritas en esta divulgación.

20 La FIG. 13 es un diagrama de bloques que ilustra un decodificador de vídeo ejemplar que puede implementar una o más técnicas descritas en esta divulgación.

La FIG. 14A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

25 La FIG. 14B es un diagrama de flujo que ilustra una operación ejemplar para decodificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

30 La FIG. 15A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

La FIG. 15B es un diagrama de flujo que ilustra una operación ejemplar para decodificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

35 La FIG. 16A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

La FIG. 16B es un diagrama de flujo que ilustra una operación ejemplar para decodificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

40 La FIG. 17 es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

45 La FIG. 18 es un diagrama de flujo que ilustra una operación ejemplar para decodificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

La FIG. 19A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

50 La FIG. 19B es un diagrama de flujo que ilustra una operación ejemplar para decodificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

55 La FIG. 20A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

La FIG. 20B es un diagrama de flujo que ilustra una operación ejemplar para decodificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

60 La FIG. 21A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

La FIG. 21B es un diagrama de flujo que ilustra una operación ejemplar para decodificación de datos de vídeo de acuerdo con una técnica de esta divulgación.

65

DESCRIPCIÓN DETALLADA

[0015] Se ha propuesto el uso de modelos de movimiento afines para proporcionar una mayor compresión de datos de vídeo. Un modelo de movimiento afín para un bloque expresa la rotación del bloque en una serie de imágenes. Se puede determinar un modelo de movimiento afín de un bloque en base a vectores de movimiento de puntos de control del bloque. En algunas implementaciones, los puntos de control del bloque son las esquinas superior izquierda y superior derecha del bloque. En algunas implementaciones, los puntos de control del bloque incluyen además la esquina inferior izquierda del bloque. Un codificador de vídeo (es decir, un codificador de vídeo o un descodificador de vídeo) puede calcular vectores de movimiento de subbloques del bloque en base a los vectores de movimiento de los puntos de control del bloque.

[0016] Se han propuesto dos técnicas principales para señalar los vectores de movimiento de los puntos de control de un bloque. La primera técnica es el intermodo afín. La segunda técnica es el modo de fusión afín. En el intermodo afín, un codificador de vídeo genera una lista de candidatos de conjunto de predictores de vectores de movimiento (MVP) afines para un bloque actual. La lista de candidatos de conjunto de MVP afines es una lista de conjuntos de MVP afines. Cada conjunto de MVP afines es un conjunto de MVP correspondientes a diferentes puntos de control del bloque actual. El codificador de vídeo señala un índice que identifica a un descodificador de vídeo un conjunto de MVP afín seleccionado en la lista de candidatos de conjunto de MVP afines. Adicionalmente, el codificador de vídeo señala una diferencia de vector de movimiento (MVD) para cada uno de los puntos de control del bloque actual. El vector de movimiento de un punto de control puede ser igual a la MVD para el punto de control más el predictor del vector de movimiento para el punto de control en el conjunto de MVP afín seleccionado. El codificador de vídeo también señala un índice de referencia que identifica una imagen de referencia que el descodificador de vídeo usa con el bloque actual. El descodificador de vídeo genera la misma lista de candidatos de conjunto de MVP afines y usa el índice señalado para determinar el conjunto de MVP afín seleccionado. El descodificador de vídeo puede agregar las MVD a los vectores de movimiento del conjunto de MVP afín seleccionado para determinar el vector de movimiento de los puntos de control del bloque actual.

[0017] En el modo de fusión afín, un codificador de vídeo y un descodificador de vídeo identifican el mismo bloque de origen afín para un bloque actual. El bloque de origen afín puede ser un bloque codificado por afinidad que colinda espacialmente con el bloque actual. El codificador de vídeo y el descodificador de vídeo extrapolan los vectores de movimiento de los puntos de control del bloque actual a partir de los vectores de movimiento de los puntos de control del bloque de origen afín. Por ejemplo, el codificador de vídeo y el descodificador de vídeo pueden construir un modelo de movimiento afín que describe vectores de movimiento de localizaciones dentro del bloque actual. El modelo de movimiento afín se define mediante un conjunto de parámetros afines. El codificador de vídeo y el descodificador de vídeo pueden determinar los parámetros afines en base a los vectores de movimiento de los puntos de control del bloque actual. El codificador de vídeo y el descodificador de vídeo pueden determinar los vectores de movimiento de los puntos de control del bloque actual en base a vectores de movimiento de puntos de control del bloque de origen afín.

[0018] De acuerdo con una técnica ejemplar de esta divulgación, cuando se genera una lista de candidatos de conjunto de MVP afines en el intermodo afín, un codificador de vídeo puede incluir, en la lista de candidatos de conjunto de MVP afines para un bloque actual, un conjunto de MVP afín que especifica vectores de movimiento extrapolado de los puntos de control de un bloque de origen afín. En este ejemplo, el codificador de vídeo puede señalar un índice en la lista de candidatos de conjunto de MVP afines, MVD para cada punto de control del bloque actual y un índice de referencia. Un descodificador de vídeo puede generar la misma lista de candidatos de conjunto de MVP afines para el bloque actual. Adicionalmente, el descodificador de vídeo usa el índice en la lista de candidatos de conjunto de MVP afines para identificar un conjunto de candidatos de MVP afines seleccionado. El descodificador de vídeo puede usar a continuación las MVD y predictores de vector de movimiento del conjunto de candidatos de MVP afines seleccionado para determinar vectores de movimiento de los puntos de control del bloque actual. Además, el descodificador de vídeo puede usar los vectores de movimiento y la imagen de referencia indicados por el índice de referencia para generar un bloque predictivo para el bloque actual. El descodificador de vídeo puede usar el bloque predictivo del bloque actual para reconstruir el bloque actual. La inclusión del conjunto de MVP afín que especifica vectores de movimiento extrapolados de los puntos de control del bloque de origen afín en la lista de candidatos de conjunto de MVP afines del bloque actual puede aumentar la eficacia de codificación.

[0019] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo ejemplar 10 que puede utilizar técnicas de esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados que se van a descodificar en un momento posterior por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 por medio de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo ordenadores de escritorio, ordenadores plegables (es decir, portátiles), ordenadores de tableta, descodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", tabletas, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivo de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica. Por tanto, el dispositivo de origen 12 y el dispositivo de destino 14 pueden ser dispositivos de comunicación inalámbrica. El dispositivo de origen 12 es un dispositivo de codificación de vídeo ejemplar (es decir, un dispositivo para codificar datos de vídeo). El dispositivo de destino 14 es un dispositivo de descodificación de vídeo

ejemplar (es decir, un dispositivo para decodificar datos de vídeo).

[0020] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un medio de almacenamiento 19 configurado para almacenar datos de vídeo, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye una interfaz de entrada 26, un medio de almacenamiento 28 configurado para almacenar datos de vídeo codificados, un decodificador de vídeo 30 y un dispositivo de visualización 32. En otros ejemplos, el dispositivo de origen 12 y el dispositivo de destino 14 incluyen otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde una fuente de vídeo externa, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[0021] El sistema ilustrado 10 de la FIG. 1 es simplemente un ejemplo. Las técnicas para el procesamiento de datos de vídeo se pueden realizar mediante cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, en general, las técnicas de esta divulgación se realizan mediante un dispositivo de codificación de vídeo, las técnicas también se pueden realizar mediante un codificador/descodificador de vídeo, denominado típicamente "CÓDEC". El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación en los que el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden funcionar de forma sustancialmente simétrica, de modo que cada uno del dispositivo de origen 12 y el dispositivo de destino 14 incluye componentes de codificación y decodificación de vídeo. Por lo tanto, el sistema 10 puede soportar una transmisión de vídeo unidireccional o bidireccional entre el dispositivo de origen 12 y el dispositivo de destino 14, por ejemplo, para la transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo o videotelefonía.

[0022] La fuente de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una interfaz de suministro de vídeo para recibir datos de vídeo desde un proveedor de contenidos de vídeo. Como otra alternativa, la fuente de vídeo 18 puede generar datos basados en gráficos de ordenador como fuente de vídeo, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. El dispositivo de origen 12 puede comprender uno o más medios de almacenamiento de datos (por ejemplo, medio de almacenamiento 19) configurados para almacenar los datos de vídeo. Las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o por cable. En cada caso, el codificador de vídeo 20 puede codificar el vídeo capturado, precapturado o generado por ordenador. La interfaz de salida 22 puede enviar la información de vídeo codificada a un medio legible por ordenador 16.

[0023] La interfaz de salida 22 puede comprender diversos tipos de componentes o dispositivos. Por ejemplo, la interfaz de salida 22 puede comprender un transmisor inalámbrico, un módem, un componente de red por cable (por ejemplo, una tarjeta Ethernet) u otro componente físico. En ejemplos donde la interfaz de salida 22 comprende un receptor inalámbrico, la interfaz de salida 22 se puede configurar para recibir datos, tal como el flujo de bits, modulados de acuerdo con un estándar de comunicación celular, tal como 4G, 4G-LTE, LTE Avanzada, 5G y similares. En algunos ejemplos donde la interfaz de salida 22 comprende un receptor inalámbrico, la interfaz de salida 22 se puede configurar para recibir datos, tal como el flujo de bits, modulados de acuerdo con otros estándares inalámbricos, tal como una especificación IEEE802.11, una especificación IEEE802.15 (por ejemplo, ZigBee™), un estándar Bluetooth™ y similares. En algunos ejemplos, la circuitería de interfaz de salida 22 se puede integrar en la circuitería del codificador de vídeo 20 y/u otros componentes del dispositivo de origen 12. Por ejemplo, el codificador de vídeo 20 y la interfaz de salida 22 pueden ser partes de un sistema en un chip (SoC). El SoC también puede incluir otros componentes, tal como un microprocesador de propósito general, una unidad de procesamiento de gráficos, y así sucesivamente.

[0024] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a decodificar por medio del medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En algunos ejemplos, el medio legible por ordenador 16 puede comprender un medio de comunicación para habilitar el dispositivo de origen 12 para transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global, tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14. El dispositivo de destino 14 puede comprender uno o más medios de almacenamiento de datos configurados para almacenar datos de vídeo codificados y datos de vídeo decodificados.

[0025] En algunos ejemplos, se pueden emitir datos codificados desde la interfaz de salida 22 hasta un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos, distribuidos o de acceso local, tales 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 para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que pueda almacenar el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados desde el dispositivo de almacenamiento por medio de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor que pueda almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de archivos ejemplares incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0026] Las técnicas se pueden aplicar a la codificación de vídeo como soporte de cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo por internet, tales como la transmisión continua adaptativa dinámica por HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones o combinaciones de los ejemplos anteriores. En algunos ejemplos, el sistema 10 puede estar configurado para admitir una transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión continua de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

[0027] El medio legible por ordenador 16 puede incluir medios transitorios, tales como una radiodifusión inalámbrica o una transmisión de red por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como un disco duro, una unidad de memoria flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, por medio de transmisión por red. De forma similar, un dispositivo informático de una instalación de producción de un medio, tal como una instalación de acuñación de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y producir un disco que contiene los datos de vídeo codificados. Por lo tanto, se puede entender que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

[0028] La interfaz de entrada 26 del dispositivo de destino 14 recibe información desde el medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información de sintaxis definida por el codificador de vídeo 20 del codificador de vídeo 20, que también se usa por el descodificador de vídeo 30, que incluye elementos de sintaxis que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, grupos de imágenes (GOP). La interfaz de entrada 26 puede comprender diversos tipos de componentes o dispositivos. Por ejemplo, la interfaz de entrada 26 puede comprender un receptor inalámbrico, un módem, un componente de red por cable (por ejemplo, una tarjeta Ethernet) u otro componente físico. En ejemplos donde la interfaz de entrada 26 comprende un receptor inalámbrico, la interfaz de entrada 26 se puede configurar para recibir datos, tal como el flujo de bits, modulados de acuerdo con un estándar de comunicación celular, tal como 4G, 4G-LTE, LTE Avanzada, 5G y similares. En algunos ejemplos donde la interfaz de entrada 26 comprende un receptor inalámbrico, la interfaz de entrada 26 se puede configurar para recibir datos, tal como el flujo de bits, modulados de acuerdo con otros estándares inalámbricos, tal como una especificación IEEE802.11, una especificación IEEE802.15 (por ejemplo, ZigBee™), un estándar Bluetooth™ y similares. En algunos ejemplos, la circuitería de interfaz de entrada 26 se puede integrar en la circuitería del descodificador de vídeo 30 y/u otros componentes del dispositivo de destino 14. Por ejemplo, el descodificador de vídeo 30 y la interfaz de entrada 26 pueden ser partes de un SoC. El SoC también puede incluir otros componentes, tal como un microprocesador de propósito general, una unidad de procesamiento de gráficos, y así sucesivamente.

[0029] El medio de almacenamiento 28 se puede configurar para almacenar datos de vídeo codificados, tales como datos de vídeo codificados (por ejemplo, un flujo de bits) recibidos por la interfaz de entrada 26. El dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0030] Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden implementar como cualquiera de una variedad de circuitería de codificador adecuada, tal como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador y ejecutar las instrucciones en hardware usando uno o más

procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden incluir en uno o más codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

[0031] En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con un estándar de codificación de vídeo tal como un estándar existente o futuro. Estándares de codificación de vídeo ejemplares incluyen, pero no se limitan a, ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluyendo sus extensiones de codificación de vídeo escalable (SVC) y de codificación de vídeo multivista (MVC). Además, recientemente se ha desarrollado un nuevo estándar de codificación de vídeo, a saber, la codificación de vídeo de alta eficacia (HEVC) o ITU-T H.265, que incluye sus extensiones de codificación de contenido de rango y pantalla, extensiones de codificación de vídeo 3D (3D-HEVC) y de múltiples vistas (MV-HEVC) y extensión escalable (SHVC) por el equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC), así como el equipo de colaboración conjunta sobre el desarrollo de la extensión de codificación de vídeo 3D (JCT-3V) del grupo de expertos en codificación de vídeo (VCEG) de UIT-T y el grupo de expertos en imágenes en movimiento ISO/IEC (MPEG). Una memoria descriptiva del borrador de HEVC, y denominado HEVC WD a continuación en el presente documento, está disponible de Wang y col., "High Efficiency Video Coding (HEVC) Defect Report", equipo colaborativo conjunto sobre codificación de vídeo (JCT-VC) de ITU-T SG16, WP 3 e ISO/IEC JTC1/SC 29/WG11, 14.^a reunión, Viena, AT, 25 de julio - 2 de agosto de 2013, documento JCTVC-N1003_v1, disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip. HEVC también se publica como recomendación UIT-T H.265, serie H: sistemas audiovisuales y multimedia, infraestructura de servicios audiovisuales - codificación de vídeo en movimiento, codificación de vídeo de alta eficacia, diciembre de 2016.

[0032] ITU-T VCEG (Q6/16) e ISO/IEC MPEG (JTC1/SC29/WG11) ahora están estudiando la posible necesidad de estandarizar la tecnología de codificación de vídeo futura con una capacidad de compresión que excede significativamente la del estándar de HEVC actual (incluyendo sus extensiones actuales y las extensiones a corto plazo para la codificación de contenido de pantalla y la codificación de alto rango dinámico). Los grupos están trabajando juntos en esta actividad de exploración en un esfuerzo de colaboración conjunto conocido como el Equipo Conjunto de Exploración de Vídeo (JVET) para evaluar los diseños de tecnología de compresión propuestos por sus expertos en esta área. El JVET se reunió primero del 19 al 21 de octubre de 2015. La última versión del software de referencia, es decir, el Modelo de exploración conjunta 3 (JEM3) se puede descargar en: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-3.0/. J. Chen, E. Alshina, GJ Sullivan, J.-R. Ohm, J. Boyce, "Algorithm Description of Joint Exploration Test Model", JVET-C1001, mayo de 2016 (a continuación en el presente documento, "JVET-C1001") incluye una descripción algorítmica del modelo de prueba de exploración conjunta 3 (JEM3.0).

[0033] En HEVC y otras memorias descriptivas de codificación de vídeo, los datos de vídeo incluyen una serie de imágenes. Las imágenes también se pueden denominar "tramas". Una imagen puede incluir una o más matrices de muestra. Cada matriz de muestra respectiva de una imagen puede comprender una matriz de muestras para un componente de color respectivo. En HEVC, una imagen puede incluir tres matrices de muestras, indicadas como S_L , S_{Cb} y S_{Cr} . S_L es una matriz bidimensional (es decir, un bloque) de muestras de luma. S_{Cb} es una matriz bidimensional de muestras de croma Cb. S_{Cr} es una matriz bidimensional de muestras de croma Cr. En otros casos, una imagen puede ser monocromática y puede incluir solo una matriz de muestras de luma.

[0034] Como parte de la codificación de datos de vídeo, el codificador de vídeo 20 puede codificar imágenes de los datos de vídeo. En otras palabras, el codificador de vídeo 20 puede generar representaciones codificadas de las imágenes de los datos de vídeo. Una representación codificada de una imagen puede denominarse en el presente documento "imagen codificada" o "imagen codificada".

[0035] Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede codificar bloques de la imagen. El codificador de vídeo 20 puede incluir, en un flujo de bits, una representación codificada del bloque de vídeo. Por ejemplo, para generar una representación codificada de una imagen, el codificador de vídeo 20 puede dividir cada matriz de muestra de la imagen en bloques de árbol de codificación (CTB) y codificar los CTB. Un CTB puede ser un bloque $N \times N$ de muestras en una matriz de muestras de una imagen. En el perfil principal de HEVC, el tamaño de un CTB puede variar de 16×16 a 64×64 , aunque técnicamente se pueden admitir tamaños de CTB de 8×8 .

[0036] Una unidad de árbol de codificación (CTU) de una imagen puede comprender uno o más CTB y puede comprender estructuras de sintaxis usadas para codificar las muestras de uno o más CTB. Por ejemplo, cada CTU puede comprender un CTB de muestras de luma, dos CTB correspondientes de muestras de croma y estructuras de sintaxis usadas para codificar las muestras de los CTB. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una CTU puede comprender un solo CTB y estructuras de sintaxis usadas para codificar las muestras del CTB. Una CTU también se puede denominar "bloque en árbol" o "unidad de codificación de máximo tamaño" (LCU). En esta divulgación, una "estructura sintáctica" se puede definir como cero o más elementos de sintaxis presentes conjuntamente en un flujo de bits en un orden específico. En algunos códecs, una imagen codificada es una representación codificada que contiene todas las CTU de la imagen.

[0037] Para codificar una CTU de una imagen, el codificador de vídeo 20 puede dividir los CTB de la CTU en uno o más bloques de codificación. Un bloque de codificación es un bloque NxN de muestras. En algunos códecs, para codificar una CTU de una imagen, el codificador de vídeo 20 puede realizar de forma recursiva una división de árbol cuaternario en los bloques de árbol de codificación de una CTU para dividir los CTB en bloques de codificación; de ahí el nombre de "unidades de árbol de codificación". Una unidad de codificación (CU) puede comprender uno o más bloques de codificación y estructuras sintácticas usadas para codificar muestras de los uno o más bloques de codificación. Por ejemplo, una CU puede comprender un bloque de codificación de muestras de luma y dos bloques de codificación correspondientes de muestras de croma de una imagen que tiene una matriz de muestras de luma, una matriz de muestras Cb y una matriz de muestras Cr y estructuras sintácticas usadas para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una CU puede comprender un único bloque de codificación y estructuras sintácticas usadas para codificar las muestras del bloque de codificación.

[0038] Además, el codificador de vídeo 20 puede codificar CU de una imagen de los datos de vídeo. En algunos códecs, como parte de la codificación de una CU, el codificador de vídeo 20 puede dividir un bloque de codificación de la CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las cuales se aplica la misma predicción. Una unidad de predicción (PU) de una CU puede comprender uno o más bloques de predicción de una CU y estructuras sintácticas usadas para predecir los uno o más bloques de predicción. Por ejemplo, una PU puede comprender un bloque de predicción de muestras de luma, dos bloques de predicción correspondientes de muestras de croma y estructuras sintácticas usadas para predecir los bloques de predicción. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una PU puede comprender un único bloque de predicción y estructuras sintácticas usadas para predecir el bloque de predicción.

[0039] El codificador de vídeo 20 puede generar un bloque predictivo (por ejemplo, un bloque predictivo de luma, Cb y Cr) para un bloque de predicción (por ejemplo, bloque de predicción de luma, Cb y Cr) de una CU. El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar un bloque predictivo. Si el codificador de vídeo 20 usa intrapredicción para generar un bloque predictivo, el codificador de vídeo 20 puede generar el bloque predictivo en base a muestras descodificadas de la imagen que incluye la CU. Si el codificador de vídeo 20 usa intrapredicción para generar un bloque predictivo de una CU de una imagen actual, el codificador de vídeo 20 puede generar el bloque predictivo de la CU en base a muestras descodificadas de una imagen de referencia (es decir, una imagen diferente a la imagen actual).

[0040] En HEVC y otros códecs particulares, el codificador de vídeo 20 codifica una CU usando sólo un modo de predicción (es decir, intrapredicción o interpredicción). Por tanto, en HEVC y otros códecs particulares, el codificador de vídeo 20 puede generar bloques predictivos de una CU usando intrapredicción o el codificador de vídeo 20 puede generar bloques predictivos de la CU usando interpredicción. Cuando el codificador de vídeo 20 usa interpredicción para codificar una CU, el codificador de vídeo 20 puede dividir la CU en 2 o 4 PU, o una PU corresponde a la CU completa. Cuando dos PU están presentes en una CU, las dos PU pueden ser rectángulos de mitad de tamaño o de tamaño de dos rectángulos con un $\frac{1}{4}$ o $\frac{3}{4}$ del tamaño de la CU. En HEVC, existen ocho modos de partición para una CU codificada con modo de interpredicción, es decir, PART_2Nx2N, PART_2NxN, PART_Nx2N, PART_NxN, PART_2NxnU, PART_2NxnD, PART_nLx2N y PART_nRx2N. Cuando se intrapredice una CU, 2Nx2N y NxN son las únicas formas de PU permitidas, y dentro de cada PU se codifica un único modo de intrapredicción (mientras que el modo de predicción de croma se señala a nivel de CU).

[0041] El codificador de vídeo 20 también puede generar uno o más bloques residuales para la CU. Por ejemplo, el codificador de vídeo 20 puede generar un bloque residual de luma para la CU. Cada muestra en el bloque residual de luma de la CU indica una diferencia entre una muestra de luma en uno de los bloques de predicción de luma de la CU y una muestra correspondiente en el bloque de codificación de luma original de la CU. Además, el codificador de vídeo 20 puede generar un bloque residual de Cb para la CU. Cada muestra del bloque residual de Cb de la CU puede indicar una diferencia entre una muestra de Cb de uno de los bloques predictivos de Cb de la CU y una muestra correspondiente del bloque de codificación de Cb original de la CU. El codificador de vídeo 20 también puede generar un bloque residual de Cr para la CU. Cada muestra en el bloque residual de Cr de la CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques predictivos de Cr de la CU y una muestra correspondiente en el bloque de codificación de Cr original de la CU.

[0042] Además, el codificador de vídeo 20 puede descomponer los bloques residuales de una CU en uno o más bloques de transformada. Por ejemplo, el codificador de vídeo 20 puede usar una división de árbol cuaternario para descomponer los bloques residuales de una CU en uno o más bloques de transformada. Un bloque de transformada es un bloque rectangular (por ejemplo, cuadrado o no cuadrado) de muestras a las que se aplica la misma transformada. Una unidad de transformada (TU) de una CU puede comprender uno o más bloques de transformada. Por ejemplo, una TU puede comprender un bloque de transformada de muestras de luma, dos bloques de transformada correspondientes de muestras de croma y estructuras sintácticas usadas para transformar las muestras de bloque de transformada. Por tanto, cada TU de una CU puede tener un bloque de transformada de luma, un bloque de transformada de Cb y un bloque de transformada de Cr. El bloque de transformada de luma de la TU puede ser un

subbloque del bloque residual de luma de la CU. El bloque de transformada de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformada de Cr puede ser un subbloque del bloque residual de Cr de la CU. En imágenes monocromáticas o imágenes que tengan tres planos de color separados, una TU puede comprender un único bloque de transformada y estructuras sintácticas usadas para transformar las muestras del bloque de transformada.

[0043] El codificador de vídeo 20 puede aplicar una o más transformadas a un bloque de transformada de una TU para generar un bloque de coeficientes para la TU. Un bloque de coeficientes puede ser una matriz bidimensional de coeficientes de transformada. Un coeficiente de transformada puede ser una cantidad escalar. En algunos ejemplos, las una o más transformadas convierten el bloque de transformada de un dominio de píxeles en un dominio de frecuencia. Por tanto, en dichos ejemplos, un coeficiente de transformada puede ser una cantidad escalar que se considera que está en un dominio de frecuencia. Un nivel de coeficiente de transformada es una cantidad entera que representa un valor asociado con un índice de frecuencia bidimensional particular en un proceso de descodificación antes del ajuste a escala para el cálculo de un valor de coeficiente de transformada.

[0044] En algunos ejemplos, el codificador de vídeo 20 omite la aplicación de la transformada al bloque de transformada. En dichos ejemplos, el codificador de vídeo 20 puede tratar valores de muestra residuales y se pueden tratar de la misma manera que los coeficientes de transformada. Por tanto, en ejemplos donde el codificador de vídeo 20 omite la aplicación de las transformadas, el siguiente análisis de los coeficientes de transformada y los bloques de coeficientes puede ser aplicable a los bloques de transformada de muestras residuales.

[0045] Después de generar un bloque de coeficientes, el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación se refiere, en general, a un proceso en el que coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. En algunos ejemplos, el codificador de vídeo 20 omite la cuantificación. Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes, el codificador de vídeo 20 puede generar elementos de sintaxis que indiquen los coeficientes de transformada cuantificados. El codificador de vídeo 20 puede codificar por entropía uno o más de los elementos de sintaxis que indican los coeficientes de transformada cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar una codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos de sintaxis que indiquen los coeficientes de transformada cuantificados. Por tanto, un bloque codificado (por ejemplo, una CU codificada) puede incluir los elementos de sintaxis codificados por entropía que indican los coeficientes de transformada cuantificados.

[0046] El codificador de vídeo 20 puede emitir un flujo de bits que incluye datos de vídeo codificados. En otras palabras, el descodificador de vídeo 20 puede emitir un flujo de bits que incluye una representación codificada de datos de vídeo. Por ejemplo, el flujo de bits puede comprender una secuencia de bits que forma una representación de imágenes codificadas de los datos de vídeo y de los datos asociados. En algunos ejemplos, una representación de una imagen codificada puede incluir representaciones codificadas de bloques.

[0047] El flujo de bits puede comprender una secuencia de unidades de capa de abstracción de red (NAL). Una unidad de NAL es una estructura sintáctica que contiene una indicación del tipo de datos en la unidad de NAL y bytes que contienen esos datos en forma de una carga útil de secuencia de byte sin procesar (RBSP) entremezclados como sea necesario con bits de prevención de emulación. Cada una de las unidades de NAL puede incluir una cabecera de unidad de NAL y encapsula una RBSP. La cabecera de unidad de NAL puede incluir un elemento de sintaxis que indique un código de tipo de unidad de NAL. El código de tipo de unidad de NAL especificado por la cabecera de unidad de NAL de una unidad de NAL indica el tipo de la unidad de NAL. Una RBSP puede ser una estructura sintáctica que contenga un número entero de bytes que esté encapsulado dentro de una unidad de NAL. En algunos casos, una RBSP incluye cero bits.

[0048] El descodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Como se indica anteriormente, el flujo de bits puede comprender una representación codificada de datos de vídeo. El descodificador de vídeo 30 puede descodificar el flujo de bits para reconstruir imágenes de los datos de vídeo. Como parte de la descodificación del flujo de bits, el descodificador de vídeo 30 puede analizar el flujo de bits para obtener elementos de sintaxis a partir del flujo de bits. El descodificador de vídeo 30 puede reconstruir imágenes de los datos de vídeo en base a, al menos en parte, los elementos de sintaxis obtenidos a partir del flujo de bits. El proceso para reconstruir imágenes de los datos de vídeo puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20 para codificar las imágenes. Por ejemplo, el descodificador de vídeo 30 puede usar interpolación o intrapredicción para generar uno o más bloques predictivos para cada PU de la CU actual; puede usar vectores de movimiento de las PU para determinar bloques predictivos para las PU de una CU actual. Además, el descodificador de vídeo 30 puede realizar una cuantificación inversa de los bloques de coeficientes de TU de la CU actual. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes para reconstruir los bloques de transformada de las TU de la CU actual. En algunos ejemplos, el descodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a las muestras descodificadas correspondientes de los bloques de transformada de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el descodificador de vídeo 30 puede reconstruir la imagen.

[0049] Un fragmento de una imagen puede incluir un número entero de CTU de la imagen. Las CTU de un fragmento se pueden ordenar consecutivamente en un orden de escaneo, tal como un orden de escaneo de trama. En HEVC, un fragmento se define como un número entero de CTU contenidas en un segmento de fragmento independiente y todos los segmentos de fragmento dependientes subsiguientes (si los hubiera) que preceden al siguiente segmento de fragmento independiente (si lo hubiera) dentro de la misma unidad de acceso. Además, en HEVC, un segmento de fragmento se define como un número entero de unidades de árbol de codificación ordenadas consecutivamente en un escaneo en mosaico y contenidas en una unidad de NAL independiente. Un escaneo en mosaico es un orden secuencial específico de CTB que divide una imagen en el que los CTB se ordenan consecutivamente en un escaneo de trama de CTB en un mosaico, mientras que los mosaicos en una imagen se ordenan consecutivamente en un escaneo de trama de los mosaicos de la imagen. Un mosaico es una región rectangular de CTB dentro de una columna de mosaicos en particular y una fila de mosaicos en particular en una imagen. Una cabecera de segmento de fragmento es una parte de un segmento de fragmento codificado que contiene los elementos de datos pertenecientes a la primera o todas las unidades de árbol de codificación representadas en el segmento de fragmento. El término "cabecera de fragmento" se aplica a una cabecera de segmento de fragmento del segmento de fragmento independiente que es un segmento de fragmento actual o el segmento de fragmento independiente más reciente que precede a un segmento de fragmento dependiente actual en orden de descodificación.

[0050] Como se menciona en resumen anteriormente, en HEVC, la unidad de codificación más grande en un fragmento se denomina un bloque de árbol de codificación (CTB) o unidad de árbol de codificación (CTU). Un CTB contiene un árbol cuaternario cuyos nodos son unidades de codificación. El tamaño de un CTB puede variar desde 16x16 a 64x64 en el perfil principal de HEVC (aunque, técnicamente, se pueden admitir tamaños de CTB de 8x8). Una unidad de codificación (CU) podría ser del mismo tamaño que un CTB y tan pequeña como 8x8. Cada unidad de codificación está codificada con un modo. Cuando una CU está intercodificada, la CU se puede dividir además en 2 o 4 unidades de predicción (PU) o convertirse en una única PU cuando no se aplican particiones adicionales. Cuando dos PU están presentes en una CU, las PU pueden ser rectángulos de mitad de tamaño o de tamaño de dos rectángulos con un $\frac{1}{4}$ o $\frac{3}{4}$ del tamaño de la CU. Cuando la CU está intercodificada, está presente un conjunto de información de movimiento para cada PU. Además, cada PU se codifica con un modo único de interpretación para obtener el conjunto de información de movimiento. En algunos estándares de codificación de vídeo, una CU no se divide en múltiples PU. Por lo tanto, en dichos estándares de codificación de vídeo, no hay distinción entre una PU y una CU. Por lo tanto, cuando las técnicas de esta divulgación se aplican a dichos estándares, el análisis de las PU puede ser aplicable a las CU.

[0051] Un codificador de vídeo puede realizar interpretación unidireccional o interpretación bidireccional para un bloque actual (por ejemplo, una CU o una PU). Al realizar la interpretación unidireccional para el bloque actual, el codificador de vídeo usa un vector de movimiento para determinar una localización en una imagen de referencia. El codificador de vídeo puede generar a continuación un bloque predictivo para el bloque actual. El bloque predictivo puede comprender un bloque de muestras en la imagen de referencia en la localización indicada por el vector de movimiento, o un bloque de muestras interpoladas de muestras de la imagen de referencia. Cuando se realiza la interpretación bidireccional, el codificador de vídeo puede realizar este proceso con una segunda imagen de referencia y un segundo vector de movimiento, generando de este modo un segundo bloque predictivo para el bloque actual. En la interpretación bidireccional, los bloques predictivos generados a partir de imágenes de referencia únicas se pueden denominar en el presente documento bloques predictivos preliminares. Además, en la interpretación bidireccional, el codificador de vídeo puede generar, en base a los dos bloques preliminares, un bloque predictivo final para el bloque actual. En algunos ejemplos, el codificador de vídeo puede generar el bloque predictivo final de modo que cada muestra en el bloque predictivo final sea un promedio ponderado de las muestras correspondientes en los bloques predictivos preliminares.

[0052] Para admitir interpretación en una imagen, un codificador de vídeo genera dos listas de imágenes de referencia para la imagen. Las listas de imágenes de referencia de la imagen incluyen imágenes de referencia que están disponibles para su uso al realizar intrapredicción de bloques en la imagen. Las dos listas de imágenes de referencia se denominan normalmente Lista 0 y Lista 1. En un ejemplo, cada imagen de referencia en la Lista 0 de la imagen aparece antes de la imagen en el orden de salida. En este ejemplo, cada imagen de referencia en la Lista 1 de la imagen aparece después de la imagen en el orden de salida. Por tanto, el uso de una imagen de referencia en la Lista 0 se puede considerar una primera dirección de interpretación y el uso de una imagen de referencia en la Lista 1 se puede considerar una segunda dirección de interpretación. El codificador de vídeo 20 y el descodificador de vídeo 30 generan la Lista 0 de la imagen con imágenes de referencia en el mismo orden. Asimismo, el codificador de vídeo 20 y el descodificador de vídeo 30 generan la Lista 1 de la imagen con imágenes de referencia en el mismo orden. Por tanto, el codificador de vídeo 20 puede indicar al descodificador de vídeo 30 una imagen de referencia en una lista de imágenes de referencia señalizando un índice de referencia que indica una localización en la lista de imágenes de referencia de la imagen de referencia.

[0053] El estándar HEVC proporciona múltiples modos de interpretación, incluyendo el modo de fusión y el modo de predicción de vector de movimiento avanzado (AMVP). En el modo de fusión, el codificador de vídeo 20 y el descodificador de vídeo 30 generan listas de candidatos de vectores de movimiento de fusión (MV) coincidentes para una PU. La lista de candidatos de MV de fusión para la PU incluye uno o más candidatos de fusión, que también

pueden denominarse predictores de vector de movimiento (MVP). En HEVC, la lista de candidatos de MV de fusión contiene hasta 5 candidatos de fusión. Cada candidato de fusión respectivo en la lista de candidatos de MV de fusión especifica uno o más vectores de movimiento y uno o más índices de referencia. Por ejemplo, un candidato de fusión puede especificar un vector de movimiento de la Lista 0 y/o un vector de movimiento de la Lista 1, y puede especificar un índice de referencia de la Lista 0 y/o un índice de referencia de la Lista 1. Un vector de movimiento de la Lista 0 es un vector de movimiento que indica una localización en una imagen de referencia en la Lista 0. Un vector de movimiento de la Lista 1 es un vector de movimiento que indica una localización en una imagen de referencia en la Lista 1. El codificador de vídeo 20 puede señalar un índice de fusión que indica una localización en la lista de candidatos de MV de fusión de un candidato de fusión seleccionado para la PU. El descodificador de vídeo 30 puede usar el índice de fusión para identificar el candidato de fusión seleccionado. El descodificador de vídeo 30 puede usar a continuación los vectores de movimiento y los índices de referencia del candidato de fusión seleccionado como vectores de movimiento e índices de referencia de la PU.

[0054] En el modo de AMVP, el codificador de vídeo 20 genera una lista de candidatos de AMVP de Lista 0 y/o una lista de candidatos de AMVP de Lista 1 para una PU, cualquiera de las cuales puede denominarse lista de candidatos de AMVP. El descodificador de vídeo 30 genera listas de candidatos de AMVP que coinciden con las listas de candidatos de AMVP generadas por el codificador de vídeo 20. En HEVC, una lista de candidatos de AMVP contiene dos candidatos de AMVP. Cada candidato de AMVP respectivo en una lista de candidatos de AMVP de Lista 0 especifica un vector de movimiento de Lista 0 respectivo. Cada candidato de AMVP respectivo en una lista de candidatos de AMVP de Lista 1 especifica un vector de movimiento de Lista 1 respectivo. En el modo de AMVP, si la PU se interpredice unidireccionalmente a partir de la Lista 0 o se interpredice bidireccionalmente, el codificador de vídeo 20 señala un índice de MVP de Lista 0, un índice de referencia de Lista 0 y una diferencia de vector de movimiento (MVD) de Lista 0. El índice de MVP de Lista 0 especifica una localización de un candidato de AMVP seleccionado en la lista de candidatos de AMVP de Lista 0. El índice de referencia de Lista 0 especifica una localización de una imagen de referencia de Lista 0 seleccionada. La MVD de Lista 0 especifica una diferencia entre un vector de movimiento de Lista 0 de la PU y el vector de movimiento de Lista 0 especificado por el candidato de AMVP seleccionado en la lista de candidatos de AMVP de Lista 0. En consecuencia, el descodificador de vídeo 30 puede usar el índice de MVP de Lista 0 y la MVD de Lista 0 para determinar el vector de movimiento de Lista 0 de la PU. El descodificador de vídeo 30 puede determinar a continuación un bloque predictivo preliminar o final para la PU que comprende muestras correspondientes a una localización en la imagen de referencia de Lista 0 seleccionada identificada por el vector de movimiento de Lista 0 de la PU. El codificador de vídeo 20 puede señalar elementos de sintaxis similares para la Lista 1 y el descodificador de vídeo 30 puede usar los elementos de sintaxis para la Lista 1 de una manera similar.

[0055] Como se puede observar anteriormente, un candidato de fusión corresponde a un conjunto completo de información de movimiento, mientras que un candidato de AMVP contiene solo un vector de movimiento para una dirección de predicción específica. Los candidatos para ambos modos de fusión y modo de AMVP se pueden obtener de forma similar a partir de los mismos bloques vecinos espaciales y temporales.

[0056] Los candidatos de MV espaciales se obtienen de los bloques vecinos que se muestran en la FIG. 2A y la FIG. 2B, para una PU específica (PUo), aunque los procedimientos para generar los candidatos a partir de los bloques difieren para los modos de fusión y de AMVP. La FIG. 2A ilustra candidatos de MV vecinos espaciales para el modo de fusión. En la modalidad de fusión, se pueden obtener hasta cuatro candidatos de MV espaciales con el orden mostrado en la FIG. 2A con números, y el orden es el siguiente: izquierda (0), arriba (1), arriba derecha (2), abajo izquierda (3) y arriba izquierda (4).

[0057] La FIG. 2B ilustra candidatos de MV vecinos espaciales para el modo de AMVP. En el modo de AMVP, los bloques vecinos se dividen en dos grupos: el grupo izquierdo consiste en los bloques 0 y 1, y un grupo superior consiste en los bloques 2, 3 y 4, como se muestra en la FIG. 2B. Para cada grupo, el candidato potencial en un bloque vecino, que se refiere a la misma imagen de referencia que la indicada por el índice de referencia señalado, tiene la prioridad más alta para ser elegido para formar un candidato final del grupo. Por ejemplo, como parte de la generación de una lista de candidatos de AMVP de Lista 0, el codificador de vídeo comprueba si el bloque 0 se predice a partir de la Lista 0 y, de ser así, si una imagen de referencia de Lista 0 del bloque 0 es la misma que la imagen de referencia de Lista 0 de la PU actual. Si el bloque 0 se predice a partir de la Lista 0 y la imagen de referencia de Lista 0 del bloque 0 es la misma que la imagen de referencia de Lista 0 de la PU actual, el codificador de vídeo incluye el vector de movimiento de Lista 0 del bloque 0 en la lista de candidatos de AMVP de Lista 0. De lo contrario, el codificador de vídeo comprueba si el bloque 0 se predice a partir de la Lista 1 y, de ser así, si la imagen de referencia de Lista 1 del bloque 0 es la misma que la imagen de referencia de Lista 0 de la PU actual. Si el bloque 0 se predice a partir de la Lista 0 y la imagen de referencia de Lista 1 del bloque 0 es la misma que la imagen de referencia de Lista 0 de la PU actual, el codificador de vídeo incluye el vector de movimiento de Lista 1 del bloque 0 en la lista de candidatos de AMVP de Lista 0. Si la imagen de referencia de Lista 1 del bloque 0 no es la misma que la imagen de referencia de Lista 0 de la PU actual, el codificador de vídeo repite este proceso con el bloque 1 en lugar del bloque 0.

[0058] Sin embargo, si el bloque 1 no se predice a partir de la Lista 1 o la imagen de referencia de Lista 1 del bloque 1 no es la misma que la imagen de referencia de Lista 0 de la PU actual, el codificador de vídeo determina si el bloque 0 se predice a partir de la Lista 0 y, de ser así, determina si la imagen de referencia de Lista 0 del bloque 0 y la imagen

de referencia de Lista 0 de la PU actual son ambas imágenes de referencia a largo plazo o ambas imágenes de referencia a corto plazo. Si la imagen de referencia de Lista 0 del bloque 0 y la imagen de referencia de Lista 0 de la PU actual son imágenes de referencia a largo plazo o la imagen de referencia de Lista 0 del bloque 0 y la imagen de referencia de Lista 0 de la PU actual son imágenes de referencia a corto plazo, el codificador de vídeo puede escalar el vector de movimiento de Lista 0 del bloque 0 basado en una diferencia temporal entre la imagen de referencia de Lista 0 del bloque 0 y la imagen de referencia de Lista 0 de la PU actual. El codificador de vídeo incluye el vector de movimiento de Lista 0 escalado en la lista de candidatos de AMVP de Lista 0. Si la imagen de referencia de Lista 0 del bloque 0 es una imagen de referencia a largo plazo y la imagen de referencia de Lista 0 de la PU actual es una imagen de referencia a corto plazo, o viceversa, el codificador de vídeo determina si el bloque 0 se predice a partir de la Lista 1 y, de ser así, determina si la imagen de referencia de Lista 1 del bloque 0 y la imagen de referencia de Lista 0 de la PU actual son ambas imágenes de referencia a largo plazo o ambas imágenes de referencia a corto plazo. Si la imagen de referencia de Lista 1 del bloque 0 y la imagen de referencia de Lista 0 de la PU actual son imágenes de referencia a largo plazo o la imagen de referencia de Lista 1 del bloque 0 y la imagen de referencia de Lista 0 de la PU actual son imágenes de referencia a corto plazo, el codificador de vídeo puede escalar el vector de movimiento de Lista 1 del bloque 0 basado en una diferencia temporal entre la imagen de referencia de Lista 1 del bloque 0 y la imagen de referencia de Lista 0 de la PU actual. El codificador de vídeo incluye el vector de movimiento de Lista 0 escalado en la lista de candidatos de AMVP de Lista 0. Si la imagen de referencia de Lista 1 del bloque 0 es una imagen de referencia a largo plazo y la imagen de referencia de Lista 0 de la PU actual es una imagen de referencia a corto plazo, o viceversa, el codificador de vídeo repite este proceso con el bloque 1 en lugar del bloque 0.

[0059] El codificador de vídeo puede realizar un proceso similar para los bloques 2, 3 y 4 para incluir un segundo candidato en la lista de candidatos de AMVP de Lista 0 de la PU actual. Adicionalmente, el codificador de vídeo puede repetir todo este proceso, intercambiando referencias a la Lista 0 con la Lista 1 y la referencia a la Lista 1 con la Lista 0, para generar la lista de candidatos de AMVP de Lista 1 de la PU actual.

[0060] Por tanto, en el modo de AVMP, los bloques vecinos se dividen en dos grupos: el grupo izquierdo que consiste en los bloques 0 y 1, y un grupo superior que consiste en los bloques 2, 3 y 4, como se muestra en la FIG. 2B. Para cada grupo, el candidato potencial en un bloque vecino, que se refiere a la misma imagen de referencia que la indicada por el índice de referencia señalado, tiene la prioridad más alta para ser elegido para formar un candidato final del grupo. Es posible que todos los bloques vecinos no contengan un vector de movimiento que apunte a la misma imagen de referencia. Por lo tanto, si no se puede encontrar dicho candidato, el primer candidato disponible se puede ajustar a escala para formar el candidato final; por tanto, las diferencias de distancia temporal se pueden compensar.

[0061] Un codificador de vídeo puede incluir un candidato de predictor de vector de movimiento temporal (TMVP), si está habilitado y disponible, en una lista de candidatos de MV de fusión después de candidatos de vector de movimiento espacial o una lista de candidatos de AMVP. Por ejemplo, en el caso de AMVP, el codificador de vídeo puede incluir el candidato de TMVP en la lista de candidatos de AMVP si los bloques vecinos espaciales no están disponibles (por ejemplo, porque los bloques vecinos espaciales están fuera de una imagen, fragmento o límite de mosaico, porque los bloques vecinos espaciales se intrapredicen, etc.). En el modo de fusión, un candidato de TMVP puede especificar vectores de movimiento de Lista 0 y/o Lista 1 de un bloque vecino temporal. Los índices de referencia para el candidato de TMVP en el modo de fusión siempre se establecen en 0. En el modo de AMVP, un candidato de TMVP especifica un vector de movimiento de Lista 0 de un bloque vecino temporal o un vector de movimiento de Lista 1 del bloque vecino temporal. El bloque vecino temporal es un bloque en una imagen de referencia. El proceso de derivación de vectores de movimiento para un candidato de TMVP puede ser el mismo tanto para el modo de fusión como de AMVP.

[0062] La FIG. 3A es un diagrama conceptual que ilustra una técnica ejemplar para obtener un candidato de TMVP. Como se muestra en la FIG. 3A, una localización de bloque principal para la derivación de candidatos de TMVP es un bloque inferior derecho 300 fuera de la PU colocada. El bloque vecino temporal a partir del cual el codificador de vídeo deriva el candidato de TMVP se coloca junto con el bloque inferior derecho 300. En el ejemplo de la FIG. 3A, el bloque inferior derecho 300 se marca como bloque "T" para indicar "temporal". El codificador de vídeo usa el bloque inferior derecho 300 en lugar de los bloques superiores o izquierdos para compensar el sesgo hacia los bloques superiores e izquierdos usados para generar candidatos vecinos espaciales. Si el bloque inferior derecho 300 está localizado fuera de la fila del CTB actual o la información de movimiento no está disponible (por ejemplo, porque un bloque de referencia temporal colocado con el bloque inferior derecho 300 se intrapredice), el bloque inferior derecho 300 se sustituye por un bloque central 302 de la PU.

[0063] Un vector de movimiento para un candidato de TMVP se obtiene a partir de la PU colocada de una denominada "imagen colocalizada". La imagen colocalizada se puede indicar en un nivel de fragmento (por ejemplo, usando un elemento de sintaxis `collocated_ref_idx`). El vector de movimiento para la PU colocalizada se denomina MV colocalizado. De forma similar al modo temporal directo en H.264/AVC, para derivar el vector de movimiento de candidatos de TMVP, el MV colocalizado se puede ajustar a escala para compensar las diferencias de distancia temporal, como se muestra en la FIG. 3B. En particular, en la FIG. 3B, cuando se codifica un bloque 320 actual de una imagen actual 322, un codificador de vídeo determina un bloque colocalizado 323 en una imagen colocalizada 324. Un vector de movimiento 326 del bloque colocalizado 323 (es decir, un vector de movimiento colocalizado) indica una localización en una imagen de referencia colocalizada 328. El codificador de vídeo genera un TMVP 330 ajustando

a escala el vector de movimiento 326 en base a una diferencia entre una distancia temporal colocalizada y una distancia temporal actual. La distancia temporal colocalizada es una distancia temporal entre la imagen colocalizada 324 y la imagen de referencia colocalizada 328. La distancia temporal actual es una distancia temporal entre la imagen actual 322 y una imagen de referencia actual 332.

[0064] Como se indica anteriormente, un codificador de vídeo puede ajustar a escala un vector de movimiento. Al ajustar a escala un vector de movimiento, se supone que el valor de un vector de movimiento es proporcional a la distancia de las imágenes en el tiempo de presentación. Un vector de movimiento asocia dos imágenes: la imagen de referencia y la imagen que contiene el vector de movimiento (a saber, la imagen contenedora). Cuando se usa un vector de movimiento para predecir otro vector de movimiento, la distancia de la imagen contenedora y la imagen de referencia se calcula en base a los valores del recuento de orden de imágenes (POC) de la imagen de referencia y la imagen contenedora.

[0065] Para que un vector de movimiento sea predicho, tanto su imagen contenedora asociada como la imagen de referencia pueden ser diferentes. Por lo tanto, se calcula una nueva distancia (basada en el POC). El vector de movimiento se ajusta a escala en base a estas dos distancias de POC. Para un candidato vecino espacial, las imágenes contenedoras para los dos vectores de movimiento son las mismas, mientras que las imágenes de referencia son diferentes. En la HEVC, el ajuste a escala del vector de movimiento se aplica tanto a la TMVP como a la AMVP para candidatos vecinos espaciales y temporales.

[0066] Además, en algunas implementaciones, si una lista de candidatos de MV (por ejemplo, una lista de candidatos de MV de fusión o una lista de candidatos de AMVP) no está completa, un codificador de vídeo puede generar e insertar candidatos de vector de movimiento artificial al final de la lista de candidatos de MV hasta que la lista de candidatos de MV tenga el número requerido de candidatos. En el modo de fusión, existen dos tipos de candidatos de MV artificiales: candidatos combinados derivados solo para fragmentos B y candidatos cero. Un candidato combinado especifica una combinación de un vector de movimiento de Lista 0 a partir de un candidato de fusión y un vector de movimiento de Lista 1 para un candidato de fusión diferente. Los candidatos cero se usan para la predicción de vectores de movimiento solo si el primer tipo (es decir, candidatos combinados) no proporciona suficientes candidatos artificiales. Un candidato cero es un candidato que especifica un MV cuyos componentes horizontal y vertical son cada uno igual a 0.

[0067] Para cada par de candidatos que ya están en la lista de candidatos y tienen la información de movimiento necesaria, los candidatos de vector de movimiento combinados bidireccionales se derivan por una combinación del vector de movimiento del primer candidato que se refiere a una imagen en la lista 0 y el vector de movimiento de un segundo candidato que se refiere a una imagen en la lista 1.

[0068] Adicionalmente, un codificador de vídeo puede aplicar un proceso de depuración para la inserción de candidatos. Los candidatos de diferentes bloques pueden ser los mismos, lo que disminuye la eficacia de una lista de candidatos de fusión/AMVP. Se aplica un proceso de depuración para resolver este problema. Compara a un candidato con los otros en la lista de candidatos actuales para evitar la inserción de un candidato idéntico en determinadas circunstancias. Para reducir la complejidad, solo se puede aplicar un número limitado de procesos de depuración, en lugar de comparar cada candidato potencial con todos los otros existentes.

[0069] En HEVC, solo se aplica un modelo de movimiento de traslación para la predicción de compensación de movimiento (MCP). Sin embargo, en el mundo real, existen muchos tipos de movimientos, por ejemplo, acercar/alejar, rotación, movimientos de perspectiva y otros movimientos irregulares. En JEM, se aplica una predicción de compensación de movimiento de transformada afín simplificada para mejorar la eficacia de codificación. Si un bloque sigue un modelo de movimiento afín, un MV de posición (x, y) en el bloque se puede determinar mediante el siguiente modelo de movimiento afín:

$$\begin{cases} v_x = ax + by + c \\ v_y = dx + ey + f \end{cases} \quad (1)$$

[0070] En la ecuación (1), v_x es un componente horizontal de un vector de movimiento para la posición (x, y) dentro del bloque, y v_y es un componente vertical del vector de movimiento para la posición (x, y) dentro del bloque. En la ecuación (1), a, b, c, d, e y f son parámetros. Debe observarse que, en el modelo de movimiento afín, diferentes posiciones dentro del bloque tienen diferentes vectores de movimiento.

[0071] En JEM3.0, el modelo de movimiento afín se simplifica a un modelo de movimiento afín de 4 parámetros asumiendo $a = e$ y $b = -d$. Por tanto, la ecuación (1) se puede simplificar como se muestra en la ecuación (1'), a continuación:

$$\begin{cases} v_x = ex - dy + c \\ v_y = dx + ey + f \end{cases} \quad (1')$$

[0072] El modelo de movimiento afín de 4 parámetros se puede representar mediante un vector de movimiento de un punto de control superior izquierdo (V_0) y un vector de movimiento de un punto de control superior derecho (V_1). La FIG. 4 ilustra un modelo de movimiento afín simplificado para un bloque actual 400. Como se muestra en la FIG. 4, un

5 campo de movimiento afín del bloque se describe mediante dos vectores de movimiento de puntos de control \widehat{V}_0 y \widehat{V}_1 . \widehat{V}_0 es un vector de movimiento de punto de control para un punto de control superior izquierdo 402 del bloque actual 400. \widehat{V}_1 es un vector de movimiento de punto de control para un punto de control superior derecho 404 del bloque actual 400.

10 **[0073]** El campo de vector de movimiento (MVF) de un bloque se describe mediante la siguiente ecuación:

$$\begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w} x - \frac{(v_{1y} - v_{0y})}{w} y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w} x + \frac{(v_{1x} - v_{0x})}{w} y + v_{0y} \end{cases} \quad (2)$$

15 **[0074]** En la ecuación (2), v_x es un componente horizontal de un vector de movimiento para una posición (x, y) en un bloque; v_y es un componente vertical del vector de movimiento para la posición (x, y) en el bloque; (v_{0x} , v_{0y}) es un vector de movimiento del punto de control de esquina superior izquierdo (por ejemplo, el punto de control superior izquierdo 402); (v_{1x} , v_{1y}) es un vector de movimiento del punto de control de esquina superior derecha (por ejemplo, punto de control superior derecho 404); y w es una anchura del bloque. Por tanto, un codificador de vídeo puede usar la ecuación (2) para "extrapolar" vectores de movimiento para posiciones (x, y) en base a vectores de movimiento de
20 puntos de control del bloque.

[0075] Para simplificar aún más la predicción de compensación de movimiento, se aplica la predicción de transformada afín basada en bloques. Por tanto, en lugar de derivar vectores de movimiento para cada localización en un bloque, un codificador de vídeo puede derivar vectores de movimiento para subbloques del bloque. En JEM, los subbloques son bloques de 4x4. Para derivar un vector de movimiento de un subbloque, el codificador de vídeo puede calcular el vector de movimiento de una muestra central del subbloque de acuerdo con la ecuación (2). El codificador de vídeo puede redondear a continuación el vector de movimiento calculado a una exactitud de fracción de 1/16. El vector de movimiento redondeado se puede denominar en el presente documento vector de movimiento de alta precisión. A continuación, el codificador de vídeo puede aplicar filtros de interpolación de compensación de movimiento para generar predicciones (es decir, bloques predictivos) de cada uno de los subbloques con vectores de movimiento
30 derivados.

[0076] La FIG. 5 ilustra un campo de vector de movimiento (MVF) afín ejemplar por subbloque. Como se muestra en el ejemplo de la FIG. 5, un bloque actual 500 tiene un punto de control superior izquierdo 502 y un punto de control superior derecho 504. Un codificador de vídeo puede calcular, en base a un vector de movimiento 506 para el punto de control superior izquierdo 502 y un vector de movimiento 508 para el punto de control superior derecho 504, vectores de movimiento para subbloques del bloque actual 500. La FIG. 5 muestra los vectores de movimiento de los subbloques como flechas pequeñas.

40 **[0077]** Después de MCP, el vector de movimiento de alta precisión de cada subbloque se redondea y guarda con la misma exactitud que el vector de movimiento normal. En algunos ejemplos, el redondeo del vector de movimiento de alta precisión solo se realiza cuando la exactitud de los vectores de movimiento almacenados es menor que la de los vectores de movimiento de alta precisión.

45 **[0078]** Existen dos modos de movimiento afines en JEM: modo AF_INTER y modo AF_MERGE. En JEM, el modo AF_INTER se puede aplicar para CU con una anchura y una altura superiores a 8. Un indicador afín se señala en el nivel de CU en el flujo de bits para indicar si se usa el modo AF_INTER. En el modo AF_INTER, el codificador de vídeo 20 señala un índice de referencia de Lista 0 y/o un índice de referencia de Lista 1 para el bloque actual para indicar una imagen de referencia de Lista 0 y/o una imagen de referencia de Lista 1.

50 **[0079]** En el modo AF_INTER, el codificador de vídeo 20 y el decodificador de vídeo 30 construyen cada uno una o más listas de candidatos (es decir, listas de candidatos de conjunto de MVP afines) para un bloque actual. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden construir cada uno una lista de candidatos de conjunto de MVP afines de Lista 0 y/o una lista de candidatos de conjunto de MVP afines de Lista 1. Cada una de las listas de candidatos de conjunto de MVP afines incluye un conjunto respectivo de conjuntos de MVP afines. En un modelo de movimiento afín de 4 parámetros, un conjunto de MVP afín en una lista de candidatos de conjunto de MVP afines de Lista 0 especifica dos vectores de movimiento de Lista 0 (es decir, un par de vectores de movimiento). En un modelo de movimiento afín de 4 parámetros, un conjunto de MVP afín en una lista de candidatos de conjunto de MVP afines de Lista 1 especifica dos vectores de movimiento de Lista 1.

60

[0080] Inicialmente, un codificador de vídeo (por ejemplo, el codificador de vídeo 20 o el decodificador de vídeo 30) intenta llenar una lista de candidatos de conjunto de MVP afines con pares de vectores de movimiento del tipo $\{(v_0, v_1) | v_0 = \{v_A, v_B, v_C\}, v_1 = \{v_D, v_E\}\}$ usando bloques vecinos. La FIG. 6A es un diagrama de bloques que ilustra un bloque actual 600 y bloques vecinos como se usa en el modo AF_INTER. Como se muestra en la FIG. 6A, v_0 se selecciona de los vectores de movimiento de los bloques A, B o C. El vector de movimiento del bloque vecino se ajusta a escala de acuerdo con la lista de referencia y una relación entre el POC de la referencia para el bloque vecino, el POC de la referencia para la CU actual y el POC de la CU actual. Por ejemplo, se supone que el codificador de vídeo selecciona un vector de movimiento de Lista 0 de un bloque vecino (por ejemplo, bloque A, B o C) como v_0 . En este ejemplo, el vector de movimiento de Lista 0 del bloque vecino indica una posición en la imagen de referencia del bloque vecino (es decir, la referencia para el bloque vecino). Además, en este ejemplo, el codificador de vídeo 20 puede seleccionar y señalar un índice de referencia de Lista 0 que indica una imagen de referencia para la CU actual (es decir, la referencia para la CU actual). Si la imagen de referencia del bloque vecino no es la misma que la imagen de referencia para la CU actual, el codificador de vídeo puede ajustar a escala el vector de movimiento de Lista 0 del bloque vecino en base a una diferencia entre una distancia temporal de referencia y una distancia temporal actual. La distancia temporal de referencia es una distancia temporal entre el POC de la imagen de referencia del bloque vecino y un POC de la CU actual. La distancia temporal actual es una distancia temporal entre el POC de la CU actual y un POC de la imagen de referencia para la CU actual. El codificador de vídeo puede realizar un proceso similar para un vector de movimiento de Lista 1. El enfoque para seleccionar v_1 de los bloques vecinos D y E es similar.

[0081] Si el número de candidatos en la lista de candidatos es menor que 2, la lista de candidatos se completa con pares de vectores de movimiento compuestos por la duplicación de cada uno de los candidatos de AMVP {AMVP0, AMVP0} y {AMVP1, AMVP1}. En otras palabras, un codificador de vídeo puede generar dos candidatos de AMVP de la manera descrita anteriormente. Los dos candidatos de AMVP se denominan AMVP0 y AMVP1. El codificador de vídeo incluye a continuación, en la lista de candidatos 620 de la FIG. 6B, un primer candidato de predictor de vector de movimiento afín que especifica AMVP0 como un vector de movimiento para un primer punto de control y especifica AMVP0 como un vector de movimiento para un segundo punto de control. Si el número de candidatos en la lista de candidatos siguen siendo menor que 2 después de incluir el primer candidato de MVP afín en la lista de candidatos 620, el codificador de vídeo incluye un segundo candidato de MVP afín en la lista de candidatos 620, donde el segundo candidato de MVP afín especifica AMVP1 como el vector de movimiento para el primer punto de control y especifica AMVP1 como el vector de movimiento para el segundo punto de control.

[0082] Cuando la lista de candidatos 620 es mayor que 2, el codificador de vídeo clasifica en primer lugar a los candidatos en la lista de candidatos 620 de acuerdo con la coherencia de los vectores de movimiento vecinos (similitud de los dos vectores de movimiento en un candidato par). El codificador de vídeo solo conserva los dos primeros candidatos, como se muestra en la FIG. 6B con el marcador lineal "Tamaño = 2". El codificador de vídeo 20 puede usar una verificación de coste de distorsión de velocidad para determinar qué candidato de conjunto de vectores de movimiento se selecciona como la predicción de vector de movimiento de punto de control (CPMVP) de la CU actual. El codificador de vídeo 20 puede señalar en el flujo de bits un índice que indica la posición de la CPMVP en la lista de candidatos 620. El decodificador de vídeo 30 puede obtener el índice del flujo de bits y usar el índice para determinar cuál de los candidatos en la lista de candidatos 620 es la CPMVP. Después de que se determina la CPMVP de la CU afín actual, se aplica la estimación del movimiento afín y se encuentra el vector de movimiento de punto de control (CPMV). El codificador de vídeo 20 señala en el flujo de bits una diferencia entre el CPMV y la CPMVP. En otras palabras, el codificador de vídeo 20 señala una diferencia de vector de movimiento (MVD) en el flujo de bits.

[0083] Además, tanto en HEVC como en JEM, un elemento de sintaxis de predicción entre imágenes, `inter_pred_idc`, indica si la Lista 0, la Lista 1 o ambas se usan para un bloque (por ejemplo, una CU o PU). Para cada MVP obtenido de una lista de imágenes de referencia, la imagen de referencia correspondiente se señala mediante un índice de la lista de imágenes de referencia, `ref_idx_l0/1`, y el MV (`_x;_y`) se representa mediante un índice del MVP, `mvp_l0/1_flag`, y su diferencia de MV (MVD). La sintaxis de MVD también se señala en el flujo de bits de modo que los MV se puedan reconstruir en el lado del decodificador. En otras palabras, si un bloque se predice unidireccionalmente a partir de la Lista 0 o se predice bidireccionalmente, el codificador de vídeo 20 señala un indicador `ref_idx_l0` para indicar una localización de una imagen de referencia en la Lista 0, señala un `mvp_l0_flag` para indicar una localización en una lista de candidatos de AMVP de Lista 0 de un predictor de vector de movimiento seleccionado y señala un MVD de Lista 0. Si un bloque se predice unidireccionalmente a partir de la Lista 1 o se predice bidireccionalmente, el codificador de vídeo 20 señala un indicador `ref_idx_l1` para indicar una localización de una imagen de referencia en la Lista 1, señala un `mvp_l1_flag` para indicar una localización en una lista de candidatos de AMVP de Lista 1 de un predictor de vector de movimiento seleccionado y señala un MVD de Lista 1.

[0084] Además, el codificador de vídeo 20 puede señalar un indicador (por ejemplo, `mvd_l1_zero_flag`) en una cabecera de fragmento. El indicador indica si el MVD para la segunda lista de imágenes de referencia (por ejemplo, Lista 1) es igual a cero y, por lo tanto, no está señalado en el flujo de bits. La no señalización de la MVD para la segunda lista de imágenes de referencia puede mejorar aún más la eficacia de codificación en algunas circunstancias.

[0085] Cuando se codifica una CU en modo AF_MERGE, un codificador de vídeo asigna a la CU el modelo de movimiento afín de un bloque codificado con modo afín a partir de los bloques reconstruidos vecinos válidos que aparecen primero en el orden de visita: A→B→C→D→E. La FIG. 7A muestra bloques vecinos usados cuando se

codifica un bloque actual 700 en modo AF_MERGE. Un orden de visita (es decir, un orden de selección) para los bloques vecinos es de izquierda, arriba, arriba a la derecha, izquierda-abajo a arriba-izquierda como se muestra en la FIG. 7A. Por ejemplo, si el bloque vecino B es el primer bloque vecino en el orden A→B→C→D→E que se codifica usando el modo afin, el codificador de vídeo puede usar el modelo de movimiento afin del bloque vecino B como el modelo de movimiento afin del bloque actual. Por ejemplo, en este ejemplo, para $X = 0$ y/o $X = 1$, el codificador de vídeo puede extrapolar el vector de movimiento de Lista X de un punto de control superior izquierdo del bloque vecino B para generar el vector de movimiento de Lista X de un punto de control superior izquierdo del bloque actual 700, usar un índice de referencia de Lista X del punto de control superior izquierdo del bloque vecino B como el índice de referencia de Lista X del punto de control superior izquierdo del bloque actual 700, extrapolar el vector de movimiento de Lista X de un punto de control superior derecho del bloque vecino B para generar el vector de movimiento de Lista X de un punto de control superior derecho del bloque actual 700, y usar un índice de referencia de Lista X del punto de control superior derecho del bloque vecino B como el índice de referencia de Lista X del punto de control superior derecho del bloque actual 700. En este ejemplo, el codificador de vídeo puede usar la ecuación (2), anterior, para extrapolar un vector de movimiento de un punto de control del bloque vecino B para determinar un vector de movimiento de un punto de control del bloque actual 700, usando una posición (x, y) del punto de control del bloque actual 700 como x e y en la ecuación (2).

[0086] Si el bloque vecino izquierdo superior vecino A está codificado en modo afin como se muestra en la FIG. 7B, se derivan los vectores de movimiento v_2 , v_3 y v_4 de la esquina superior izquierda, la esquina inferior derecha y la esquina inferior izquierda del bloque actual 720 que contiene el bloque A. El vector de movimiento v_0 de la esquina superior izquierda del bloque actual 720 se calcula de acuerdo con v_2 , v_3 y v_4 . En segundo lugar, se calcula el vector de movimiento v_1 de la parte superior derecha de la CU actual. Para ser específico, primero se construye un modelo de movimiento afin de 6 parámetros mediante los vectores de movimiento v_2 , v_3 y v_4 como se muestra en la ecuación (3), posterior, y los valores v_0 y v_1 se calculan a continuación mediante este modelo de movimiento afin de 6 parámetros. Es decir, cuando se usa con la posición (x, y) del punto de control superior izquierdo del bloque actual 720 (es decir, \widehat{v}_0), v_x en la ecuación (3) es el componente x del vector de movimiento de \widehat{v}_0 (es decir, v_{0x}) y x_y en la ecuación (3) es el componente y de \widehat{v}_0 (es decir, v_{0y}). De forma similar, cuando se usa con la posición (x, y) del punto de control superior derecho del bloque actual 720 (es decir, \widehat{v}_1), v_x en la ecuación (3) es el componente x del vector de movimiento de \widehat{v}_1 (es decir, v_{1x}) y x_y en la ecuación (3) es el componente y de \widehat{v}_1 (es decir, v_{1y}).

$$\begin{cases} v_x = \frac{(v_{3x} - v_{2x})}{w} x + \frac{(v_{4x} - v_{2x})}{h} y + v_{2x} \\ v_y = \frac{(v_{3y} - v_{2y})}{w} x + \frac{(v_{4y} - v_{2y})}{h} y + v_{2y} \end{cases} \quad (3)$$

[0087] Después de derivar el CPMV de las CU actuales v_0 y v_1 , de acuerdo con el modelo de movimiento afin simplificado de la ecuación (1), se genera el campo de vector de movimiento de la CU actual. Para identificar si la CU actual está codificada con el modo AF_MERGE, se señala un indicador afin en el flujo de bits cuando hay al menos un bloque vecino codificado en modo afin.

[0088] Además del modelo de movimiento afin de 4 parámetros en JEM, en JVET_C0062 se describe un modelo de movimiento afin de 6 parámetros. En el modelo afin de 6 parámetros, no hay restricciones en los factores de ajuste a escala entre las direcciones horizontal y vertical. Se usan tres vectores de movimiento de esquina para representar el modelo de 6 parámetros.

[0089] La FIG. 8A ilustra bloques ejemplares usados en un modelo de movimiento afin de 6 parámetros. La FIG. 8B ilustra una lista de candidatos de conjunto de MVP afin ejemplar 820 usada en un modelo de movimiento afin de 6 parámetros. Las siguientes seis ecuaciones describen las componentes horizontal (x) y vertical (y) de los vectores de movimiento en las tres esquinas (V_0 , V_1 y V_2 , como se ilustra en la FIG. 8A):

$$\begin{cases} V0_x = c \\ V0_y = f \end{cases}$$

$$\begin{cases} V1_x = a \times anchura + c \\ V1_y = d \times anchura + f \end{cases} \quad (4)$$

$$\begin{cases} V2_x = b \times altura + c \\ V2_y = e \times altura + f \end{cases}$$

[0090] Al resolver las ecuaciones (4), el modelo afín de 6 parámetros se puede determinar sustituyendo las soluciones en la ecuación (1).

[0091] Similar al modo AF_INTER de 4 parámetros, una lista de candidatos con un conjunto de vectores de movimiento $\{(v_0, v_1, v_2) | v_0 = \{v_A, v_B, v_C\}, v_1 = \{v_D, v_E\}, v_2 = \{v_F, v_G\}\}$ para el modo AF_INTER de 6 parámetros se construye usando los bloques vecinos. Por tanto, en el ejemplo de la FIG. 8B, cada conjunto de MVP afín (es decir, cada candidato) en la lista de candidatos de conjunto de MVP afín 820 incluye tres vectores de movimiento. En el modo AF_INTER, un codificador de vídeo puede generar dos listas de candidatos de conjunto de MVP afines del tipo mostrado en la FIG. 8B. Una de las listas de candidatos de conjunto de MVP afines incluye conjuntos de MVP afines que especifican los vectores de movimiento de la Lista 0. La otra lista de candidatos de conjunto de MVP afines incluye conjuntos de MVP afines que especifican los vectores de movimiento de la Lista 1.

[0092] El diseño de movimiento afín en HEVC/JEM puede tener los siguientes problemas. Por ejemplo, la correlación del movimiento afín de la Lista 0 y la Lista 1 no se utiliza para la predicción de MV para bloques codificados con el modo afín. En otro ejemplo, la correlación del movimiento afín de un bloque actual y un bloque vecino no se utiliza para la predicción de MV para bloques codificados con modo afín. En aún otro ejemplo, el MVD cero de la Lista 1 puede dañar el rendimiento de los modelos de movimiento afines debido a un MV inexacto de los puntos de control. Esta divulgación describe técnicas que pueden superar estos problemas y mejorar potencialmente la eficacia de la codificación.

[0093] En algunos ejemplos de esta divulgación, el modelo de movimiento afín es un modelo de movimiento de 6 parámetros como se muestra en la ecuación (1). En JEM-3.0, el modelo de movimiento afín para un bloque está representado por los vectores de movimiento de los puntos de control (v_0, v_1). En JVET-C0062, el modelo de movimiento afín para un bloque está representado por los vectores de movimiento de los puntos de control (v_0, v_1, v_2). Sin embargo, puede ser deseable representar el modelo de movimiento afín señalizando los parámetros a, b, c, d, e, f en la ecuación (1) o 4 parámetros simplificados. El modelo de movimiento afín también se puede interpretar además como la ecuación (5) donde O_x y O_y son los desplazamientos de traslación, S_x y S_y son la relación de ajuste a escala en las direcciones x e y y θ_x y θ_y son los ángulos de rotación.

$$\begin{cases} v_x = s_x * \cos \theta_x * x - s_y * \sin \theta_y * y + O_x \\ v_y = s_x * \sin \theta_x * x + s_y * \cos \theta_y * y + O_y \end{cases} \quad (5)$$

[0094] Esta divulgación propone varios procedimientos para mejorar el predictor de vector de movimiento (MVP) o la predicción de parámetros de la predicción de movimiento afín. Debe observarse que un codificador de vídeo puede realizar la predicción de parámetros para la representación de a, b, c, d, e, f en la ecuación (1) u $O_x, O_y, S_x, S_y, \theta_x$ y θ_y en la ecuación (5).

[0095] Las diversas técnicas de esta divulgación se enumeran a continuación.

- 1) Predicción de vectores de movimiento afines y predicción de parámetros afines entre la Lista 0 y la Lista 1.
- 2) Predicción de vector de movimiento afines entre puntos de control y predicción de parámetros afines entre conjuntos de parámetros.
- 3) Predicción de vectores de movimiento afines y predicción de parámetros afines a partir de bloques vecinos. Los bloques vecinos no se limitan a ser bloques vecinos espaciales. Más bien, en algunos ejemplos, se usan bloques vecinos temporales.
- 4) Predicción de movimiento afín de subbloque y predicción de parámetro afín de subbloque, en el que cada subbloque puede tener sus propios puntos de control y/o parámetros.
- 5) Refinar la generación adicional de candidatos de MVP
- 6) Deshabilitar la MVD L1 cero para el intermodo afín en fragmentos de GPB.

[0096] Las técnicas de esta divulgación se pueden aplicar individualmente. De forma alternativa, se puede aplicar cualquier combinación de las técnicas. Esta divulgación elabora más detalles de cada una de las técnicas a continuación.

[0097] Como se menciona anteriormente, una de las deficiencias del diseño de movimiento afín en HEVC y JEM es que no se utiliza la correlación entre el movimiento afín de Lista 0 y Lista 1. En otras palabras, los enfoques existentes señalizan puntos de control afines independientemente para la Lista 0 y la Lista 1. La explotación de la correlación entre el movimiento afín de Lista 0 y Lista 1 puede representar una oportunidad para aumentar la eficacia de la codificación.

[0098] Por tanto, de acuerdo con una técnica de esta divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden usar un modelo de movimiento afín en una dirección de interpredicción para mejorar la señalización del modelo de movimiento afín en otra dirección de interpredicción. Un modelo de movimiento afín en una dirección de interpredicción particular es un modelo de movimiento afín que especifica vectores de movimiento que apuntan a localizaciones en imágenes de referencia en una lista de imágenes de referencia particular correspondiente a la dirección de interpredicción.

[0099] Por ejemplo, para utilizar la correlación entre los MV de la Lista 0 y la Lista 1, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden usar los MV de la Lista 0 como un predictor de MV para los MV de la Lista 1, o viceversa. El descodificador de vídeo 30 puede decidir usar la predicción de MV de Lista 0 a Lista 1 o la predicción de MV de Lista 1 a Lista 0 de acuerdo con una indicación de dirección de predicción. La indicación de la dirección de predicción puede ser señalizada explícitamente o derivada implícitamente.

[0100] El descodificador de vídeo 30 puede derivar una indicación de dirección de predicción implícita usando información descodificada o reconstruida. En un ejemplo, la indicación de dirección de predicción depende de los MVP en las listas de candidatos del conjunto de MVP afines (es decir, listas de MVP) de cada dirección de predicción. Si una lista de MVP de una dirección de predicción contiene MVP derivados de algunos procedimientos menos preferentes (por ejemplo, un predictor de vector de movimiento ajustado a escala o un MVP derivado de un bloque codificado con un modo de compensación de iluminación local), el modelo afín de otra dirección de predicción se usa para predecir el modelo afín en la dirección de predicción actual. De forma alternativa, la indicación de la dirección de predicción puede depender de cuántos predictores de MV diferentes haya en cada lista.

[0101] En otro ejemplo, si una lista de MVP de una dirección de predicción contiene MVP derivados de algunos procedimientos de mayor preferencia (por ejemplo, un predictor de vector de movimiento no ajustado a escala, o es de un bloque codificado con el modo de conversión ascendente de frecuencia de tramas (FRUC)), el modelo afín de una dirección de predicción actual se usa para predecir el modelo afín en la otra dirección de predicción. En el modo de FRUC, la información de movimiento de un bloque no se señala, sino que se deriva en el lado de descodificación.

[0102] En un ejemplo, si hay algún MVP de baja prioridad en la lista de candidatos del conjunto de MVP para la Lista 1, la Indicación de dirección de predicción se establece como Lista 0 a Lista 1 y en el proceso de predicción de MV, los MV de Lista 0 se usan como el MVP para los MV de la Lista 1. De lo contrario, si no hay un MVP de baja prioridad en la lista de candidatos del conjunto de MVP para la Lista 1, la indicación de dirección de predicción se establece como Lista 1 a Lista 0 y en el proceso de predicción de MV, los MV de la Lista 1 se usan como los MVP para el MV de Lista 0.

[0103] Después de que se determina la indicación de dirección de predicción, los candidatos del conjunto de MVP que se van a reemplazar (por N candidatos) se determinan de acuerdo con la información descodificada o de reconstrucción en el lado del descodificador. En un ejemplo, el candidato de conjunto de MVP que se va a reemplazar es el primer candidato de conjunto de MVP que contiene al menos un MVP de baja prioridad. En un ejemplo, la baja prioridad se puede dar para los MVP espaciales ajustados a escala de acuerdo con la información de POC, los MVP de AMVP rellenos y los MVP temporales.

[0104] En un ejemplo, además, cuando no hay ningún candidato de conjunto de MVP que se va a reemplazar, no se realiza la predicción de vector de movimiento afín entre la Lista 0 y la Lista 1. En algunos ejemplos, el número de candidatos *N* del conjunto de MVP que se va a reemplazar se establece en 1. En un ejemplo, si la indicación de dirección de predicción se establece como predicción de MV de Lista 0 a Lista 1, los MVP del candidato de conjunto de MVP que se va a reemplazar en la Lista 1 son reemplazados por los MV ajustados a escala de los puntos de control correspondientes en la Lista 0, respectivamente, o viceversa.

[0105] En otro ejemplo, si la indicación de dirección de predicción se establece como predicción de MV de Lista 0 a Lista 1, solo los MVP parciales del candidato de conjunto de MVP que se va a reemplazar en la Lista 1 se reemplazan por los MV ajustados a escala de los puntos de control correspondientes en la Lista 0, respectivamente, o viceversa. Por ejemplo, solo se reemplaza el primer MVP (V0).

[0106] En otro ejemplo, si la indicación de dirección de predicción se establece como predicción de MV de Lista 0 a Lista 1, solo los primeros MVP (V0) del candidato de conjunto de MVP que se va a reemplazar en la Lista 1 se reemplazan por los MV ajustados a escala de los puntos de control correspondientes en la Lista 0 respectivamente. El segundo MVP (V1) del candidato de conjunto de MVP que se va a reemplazar en la Lista 1 se reemplaza por los MV ajustados a escala de los primeros puntos de control (V0) en la Lista 0 más la diferencia entre los MV del primer y segundo punto de control en la Lista 0 (V1-V0). El enfoque para reemplazar los terceros MVP (V2) por el modelo afín de 6 parámetros es similar. Si la indicación de dirección de predicción se establece como predicción de MV de Lista 1 a Lista 0, solo los primeros MVP (V0) del candidato de conjunto de MVP que se va a reemplazar en la Lista 0 se reemplazan por los MV ajustados a escala de los puntos de control correspondientes en la Lista 1, respectivamente. El segundo MVP (V1) del candidato de conjunto de MVP que se va a reemplazar en la Lista 0 se reemplaza por los MV ajustados a escala de los primeros puntos de control (V0) en la Lista 1 más la diferencia entre los MV del primer y segundo punto de control en la Lista 1 (V1-V0). Y el enfoque para reemplazar los terceros MVP (V2) por el modelo

afín de 6 parámetros es similar.

[0107] En algunos ejemplos, los MV de los puntos de control se pueden usar como candidato de MVP para los MV de los otros puntos de control. En un ejemplo, el MV del punto de control superior izquierdo (V0) se usa como candidato de MVP para el MV del punto de control superior derecho (V1). De forma similar, en el modelo de movimiento afín de 6 parámetros, el MV del punto de control superior izquierdo (V0) se usa como candidato de MVP para el MV del punto de control superior derecho (V1). Además, esta predicción del vector de movimiento entre puntos de control se puede aplicar de forma selectiva. De forma alternativa, en el modelo afín de 6 parámetros, el vector de movimiento del punto de control superior izquierdo (V0) se usa como predictor del vector de movimiento del punto de control superior derecho (V1) (o el punto de control inferior izquierdo (V2)), y el par de V0 y V1 (o el par de V0 y V2) se usa para derivar el predictor de vector de movimiento para V2 (o V1) usando un modelo de movimiento afín de 4 parámetros. En un ejemplo, solo al segundo candidato de conjunto de MVP se le aplica la predicción de MV entre puntos de control.

[0108] En otro ejemplo, se puede derivar un modelo afín a partir del vector de movimiento de un punto de control y parámetros de modelo señalizados adicionales. Los parámetros señalizados incluyen, pero no se limitan a, las diferencias de vector de movimiento entre puntos de control o grado de rotación afín, y así sucesivamente. En un ejemplo, un vector de movimiento de uno de los puntos de control del bloque actual, junto con el grado de rotación, se señalizan para un bloque codificado como modo de movimiento afín. Para cada bloque afín, el modelo de movimiento afín se construye usando el vector de movimiento del punto de control y el ángulo de rotación.

[0109] De acuerdo con técnicas particulares de esta divulgación, un codificador de vídeo puede usar vectores de movimiento extrapolados vecinos como predictores de vectores de movimiento afines para puntos de control de un bloque afín actual. Por ejemplo, para un interbloque actual, un codificador de vídeo puede utilizar modelos de movimiento de uno o más de los bloques afines vecinos del interbloque actual (por ejemplo, bloques vecinos codificados en modo AF_INTER o modo AF_MERGE), denominados bloques afines de origen, para predecir un modelo de movimiento afín para el interbloque actual (es decir, el modelo de movimiento afín actual). Por ejemplo, los MVP de los puntos de control del bloque actual se pueden extrapolar a partir de los puntos de control de los bloques vecinos. Por ejemplo, para cada punto de control respectivo del bloque actual, el codificador de vídeo puede usar la ecuación (2) anterior, para extrapolar un MVP para el punto de control respectivo del bloque actual a partir de los vectores de movimiento de los puntos de control de un bloque afín de origen. Los bloques afines de origen pueden ser uno o más bloques vecinos espaciales o bloques vecinos temporales.

[0110] En un ejemplo, el bloque afín de origen se determina como el primer bloque codificado con modo afín a partir de los bloques vecinos espaciales válidos en base a un orden de visita predefinido (por ejemplo, $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ o $\beta \rightarrow A \rightarrow D \rightarrow C \rightarrow E$ o cualquier otro orden de visita de los bloques mostrados en la FIG. 7A).

[0111] En otro ejemplo, el bloque afín de origen se determina como el primer bloque codificado con modo afín de los bloques vecinos de acuerdo con uno o más conjuntos de prioridades predefinidos basados en un orden de visita predefinido (por ejemplo, $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ o $B \rightarrow A \rightarrow D \rightarrow C \rightarrow E$ o cualquier otro orden de visita como se muestra en la FIG. 7A y la FIG. 7B). Los bloques afines vecinos que no cumplen ninguna de las prioridades se consideran no disponibles.

[0112] En algunos ejemplos, el bloque afín de origen se determina de acuerdo con un orden de visita primero y a continuación un orden de prioridad predefinido. Por ejemplo, el bloque afín de origen se puede determinar de acuerdo con el siguiente orden: A (prioridad 1) \rightarrow B (prioridad 1) \rightarrow C (prioridad 1) \rightarrow D (prioridad 1) \rightarrow E (prioridad 1) \rightarrow A (prioridad 2) \rightarrow y así sucesivamente. En este ejemplo, un codificador de vídeo comprueba primero si el bloque A está en el conjunto de prioridad 1; en caso contrario, el codificador de vídeo comprueba si el bloque B está en el conjunto de prioridad 1; en caso contrario, el codificador de vídeo comprueba si el bloque C está en el conjunto de prioridad 1; y así sucesivamente.

[0113] En otro ejemplo, un codificador de vídeo puede determinar el bloque afín de origen de acuerdo con un orden de prioridad predefinido primero y a continuación el orden de visita. Por ejemplo, A (prioridad 1) \rightarrow A (prioridad 2) \rightarrow B (prioridad 1) \rightarrow B (prioridad 2) \rightarrow C (prioridad 1) \rightarrow C (prioridad 2) \rightarrow y así sucesivamente. Por tanto, en este ejemplo, el codificador de vídeo comprueba primero si el bloque A está en el conjunto de prioridad 1; en caso contrario, el codificador de vídeo comprueba si el bloque A está en el conjunto de prioridad 2; en caso contrario, el codificador de vídeo comprueba si el bloque B está en el conjunto de prioridad 1; y así sucesivamente.

[0114] En diversos ejemplos, los conjuntos de prioridad se definen de diferentes formas. Las definiciones de diversos conjuntos de prioridad ejemplares se enumeran a continuación. Las siguientes definiciones de conjuntos de prioridad se pueden aplicar individualmente. De forma alternativa, se puede aplicar cualquier combinación de los mismos.

[0115] Un primer conjunto de prioridad ejemplar se define a continuación, donde los números más pequeños representan una prioridad más alta:

1. Un bloque afín vecino está en el conjunto de prioridad 1 si una imagen de referencia de Lista X del bloque afín vecino es la misma imagen de referencia que una imagen de referencia de Lista X del bloque actual, donde la Lista

X es la lista de imágenes de referencia del bloque actual que está siendo evaluado y X es 0 o 1.

2. Un bloque afín vecino está en el conjunto de prioridad 2 si una imagen de referencia de Lista X del bloque afín vecino es la misma imagen de referencia que una imagen de referencia de Lista Y del bloque actual, donde la Lista Y es la lista de imágenes de referencia distinta de la lista de imágenes de referencia del bloque actual que está siendo evaluado e Y es 0 o 1.

[0116] En otro ejemplo, un conjunto de prioridad se define a continuación:

1. Un bloque afín vecino está en el conjunto de prioridad 1 si una imagen de referencia de Lista X del bloque afín vecino es distinta de una imagen de referencia que una imagen de referencia de Lista X del bloque actual, donde la Lista X es la lista de imágenes de referencia del bloque actual que está siendo evaluado y X es 0 o 1.

2. Un bloque afín vecino está en el conjunto de prioridad 2 si una imagen de referencia de Lista Y del bloque afín vecino es diferente de una imagen de referencia de Lista Y del bloque actual, donde la Lista Y es una lista de imágenes de referencia distinta de una lista de imágenes de referencia del bloque actual que está siendo evaluado e Y es 0 o 1.

[0117] En otro ejemplo, el conjunto de prioridad se define a continuación:

1. Las diferencias de MV de los bloques afines vecinos están dentro de un intervalo predefinido.

2. Las diferencias de MV de los bloques afines vecinos no están dentro de un intervalo predefinido.

[0118] En otro ejemplo, el conjunto de prioridad se define a continuación. En este y otros ejemplos, números más pequeños pueden representar una prioridad más alta.

1. Un bloque afín vecino está en el conjunto de prioridad 1 si el bloque afín vecino está codificado en modo AF_INTER.

2. Un bloque afín vecino está en el conjunto de prioridad 2 si el bloque afín vecino está codificado en modo AF_MERGE.

[0119] En otro ejemplo, la prioridad de un bloque afín vecino depende de si el bloque afín vecino tiene la misma dirección de interpredicción que el bloque afín actual. En otro ejemplo, la prioridad de un bloque afín vecino depende del tamaño del bloque afín vecino. Por ejemplo, los bloques afines vecinos con tamaños más grandes pueden tener mayor prioridad.

[0120] En un ejemplo, el codificador de vídeo selecciona, como bloque afín de origen para la Lista X, un bloque afín vecino que tiene la misma imagen de referencia de Lista X que el bloque actual y aparece primero en el siguiente orden de visita: $B \rightarrow A \rightarrow D \rightarrow C \rightarrow E$. En este ejemplo, si no hay ningún bloque afín vecino disponible (por ejemplo, ninguno de los bloques afines vecinos tiene la misma imagen de referencia de Lista X que el bloque actual), el codificador de vídeo puede seleccionar, como bloque afín de origen, el bloque afín vecino que tiene como imagen de referencia de Lista Y, la imagen de referencia de Lista X del bloque actual y aparece primero en el siguiente orden de visita: $B \rightarrow A \rightarrow D \rightarrow C \rightarrow E$, donde X es 0 o 1 e Y es (1-X).

[0121] En un ejemplo, en caso de que se use el modelo afín de bipredicción en el bloque actual, el bloque afín de origen para la Lista 0 y la Lista 1 puede ser diferente. En otras palabras, el codificador de vídeo puede usar diferentes bloques afines de origen al evaluar la Lista 0 y la Lista 1 para el bloque actual. El proceso de selección de bloques afines de origen mencionado anteriormente se puede aplicar individualmente para cada lista de imágenes de referencia.

[0122] Después de que el codificador de vídeo selecciona el bloque afín de origen, el codificador de vídeo extrapola un conjunto de predictores de MV para los puntos de control del bloque actual usando los MV de los puntos de control del bloque afín de origen. Por ejemplo, en un modelo de movimiento afín de 4 parámetros, el codificador de vídeo puede extrapolar un MV de Lista X del primer punto de control del bloque actual a partir de un MV de Lista X del primer punto de control del bloque afín de origen. Adicionalmente, en este ejemplo, el codificador de vídeo puede extrapolar un MV de Lista X del segundo punto de control del bloque actual a partir de un MV de Lista X del segundo punto de control del bloque afín de origen. En este ejemplo, X es 0 o 1 y el par resultante de MV de Lista X extrapolados se denomina conjunto de predictores de vectores de movimiento extrapolados (MVP) y se puede indicar como $\{V'_0, V'_1\}$. El codificador de vídeo puede usar la ecuación (2) para realizar la extrapolación, como se describe en otra parte de esta divulgación. En un modelo de movimiento afín de 6 parámetros, el codificador de vídeo puede extrapolar también un MV de Lista X del tercer punto de control del bloque actual a partir de un MV de Lista X del tercer punto de control del bloque afín de origen. El MV de Lista X extrapolado del tercer punto de control del bloque actual también se puede incluir en un conjunto de predictores de vector de movimiento y se puede indicar como $\{V'_0, V'_1, V'_2\}$.

[0123] El codificador de vídeo puede insertar a continuación el conjunto de MVP de la Lista X extrapolado (por ejemplo, $\{V'_0, V'_1\}$ para un modelo de movimiento afín de 4 parámetros o $\{V'_0, V'_1, V'_2\}$ para un modelo de movimiento afín de 6 parámetros) en una lista de candidatos de conjunto de MVP afines de la Lista X. Después de que el codificador de vídeo inserta el conjunto de MVP extrapolado en la lista de candidatos de conjunto de MVP afines de la Lista X, el codificador de vídeo inserta un conjunto de candidatos de MVP afines convencionales en la lista de candidatos de conjunto de MVP afines de la Lista X. El candidato de conjunto de MVP afín convencional puede ser un candidato de conjunto de MVP afín generado de acuerdo con otros ejemplos proporcionados en esta divulgación. El codificador de vídeo puede insertar el candidato de conjunto de MVP convencional en la lista de candidatos de conjunto de MVP afines de la Lista X después o antes del conjunto de MVP extrapolado. Si el bloque actual se predice bidireccionalmente, el codificador de vídeo puede realizar un proceso similar para la Lista Y, donde Y es igual a 1-X.

[0124] La FIG. 9 ilustra una lista de candidatos de conjunto de MVP afín ejemplar 900 que incluye un conjunto de MVP extrapolado 902, de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 9, el conjunto de MVP extrapolado $\{V'_i, V'_j\}$ (902) se inserta en la primera posición de la lista de candidatos del conjunto de MVP afín 900 seguida por candidatos del conjunto de MVP convencional. El codificador de vídeo puede construir el resto de la lista de candidatos de conjunto de MVP afines 900 de la misma manera que la lista de candidatos 620 (FIG. 6B) o la lista de candidatos de conjunto de MVP afines 820 (FIG. 8B).

[0125] En un ejemplo, el proceso de extrapolación puede ser el mismo que el modo AF_MERGE descrito anteriormente usando la ecuación (1) o (2), dependiendo de si se usa un modelo de movimiento afín de 4 parámetros o un modelo de movimiento afín de 6 parámetros para realizar la extrapolación de MV. De forma alternativa, se pueden aplicar otras funciones de extrapolación. Por ejemplo, un codificador de vídeo puede aplicar una función bilineal a los vectores de movimiento de los puntos de control de un bloque de origen afín para realizar el proceso de extrapolación.

[0126] En algunos ejemplos, el codificador de vídeo selecciona un segundo bloque afín de origen además de seleccionar un bloque afín de origen como se describe anteriormente. El codificador de vídeo puede seleccionar el segundo bloque afín de origen si continúa buscando un bloque afín de origen después de seleccionar el primer bloque afín de origen. El codificador de vídeo puede realizar la búsqueda de acuerdo con cualquiera de los ejemplos descritos anteriormente para seleccionar el bloque afín de origen. El codificador de vídeo puede extrapolar un segundo conjunto de MVP para los puntos de control del bloque actual usando los MV de los puntos de control del segundo bloque afín de origen y puede insertar el segundo conjunto de MVP como otro candidato de conjunto de MVP en la lista de candidatos de conjunto de MVP afines.

[0127] En otro ejemplo, un codificador de vídeo selecciona dos o más bloques afines de origen cuando codifica un bloque actual. En este ejemplo, el codificador de vídeo deriva un conjunto de predictores de MV para los puntos de control del bloque actual usando los MV de algunos o el codificador de vídeo inserta todos los puntos de control de los bloques afines de origen como otro candidato de conjunto de MVP.

[0128] De acuerdo con una técnica de esta divulgación, en el intermodo afín (es decir, AF_INTER) o modo de fusión afín (es decir, AF_MERGE), el movimiento afín de cada subbloque (por ejemplo, bloque de 4x4) de un bloque actual se puede predecir o heredar directamente del movimiento extrapolado de sus propios bloques vecinos. En un ejemplo, el bloque vecino se selecciona como el bloque afín vecino más cercano para cada subbloque. En otras palabras, un bloque actual puede dividirse en una pluralidad de subbloques de igual tamaño (por ejemplo, subbloques de 4x4). Para cada subbloque respectivo de la pluralidad de subbloques, el codificador de vídeo puede determinar un bloque más cercano que se predijo usando un modelo de movimiento afín. En los casos donde el subbloque respectivo se encuentra a lo largo de un borde del bloque actual, el bloque más cercano que se predijo usando un modelo de movimiento afín puede estar fuera del bloque actual. Por ejemplo, para un subbloque superior izquierdo del bloque actual, el bloque más cercano que se predijo usando un modelo de movimiento afín puede ser un bloque arriba y a la izquierda del subbloque superior izquierdo. De forma similar, para un subbloque superior derecho del bloque actual, el bloque más cercano que se predijo usando un modelo de movimiento afín puede ser un bloque encima del subbloque superior derecho o un bloque arriba y a la derecha del subbloque superior derecho. Para un subbloque en el interior del bloque actual, el bloque más cercano que se predijo usando un modelo de movimiento afín puede ser otro subbloque del bloque actual que está arriba o a la izquierda del subbloque.

[0129] Este enfoque difiere de la forma en que se usan los subbloques en JEM3.0. Como se analiza anteriormente, en JEM3.0, un codificador de vídeo calcula los vectores de movimiento de cada subbloque de un bloque actual en base solo a los vectores de movimiento de los puntos de control en las esquinas superior izquierda y superior derecha del bloque actual. Por el contrario, de acuerdo con esta técnica de esta divulgación, los vectores de movimiento de los subbloques no se calculan en base a los vectores de movimiento de los puntos de control, sino que se predicen o heredan directamente del movimiento extrapolado de bloques vecinos. Esto puede dar como resultado una mayor eficacia de codificación porque los vectores de movimiento de los subbloques pueden tomarse de localizaciones más cercanas a los subbloques.

[0130] La FIG. 10 ilustra predicción de movimiento de subbloque o predicción de parámetro, de acuerdo con una técnica de esta divulgación, donde el movimiento afín de cada subbloque (por ejemplo, un bloque de 4x4) de un bloque actual 1000 se puede predecir o heredar directamente del movimiento extrapolado de sus propios bloques vecinos.

Como se muestra en el ejemplo de la FIG. 10, el movimiento afín de un subbloque TL1002 se predice usando el movimiento extrapolado del bloque vecino B2 (1004) mientras que el subbloque TR 1006 usa el movimiento extrapolado del bloque vecino B1. En otro ejemplo, el subbloque también puede usar movimiento extrapolado de los bloques vecinos temporales.

[0131] En el intermodo afín o en el modo de fusión afín, los parámetros afines (por ejemplo, a, b, c, d, e y f en las ecuaciones (1) o (4) de cada subbloque (por ejemplo, el bloque de 4x4) de un bloque actual se pueden predecir o heredar directamente de sus propios bloques vecinos. En un ejemplo, el bloque vecino se selecciona como el bloque afín vecino más cercano para cada subbloque. Por ejemplo, como se muestra en la FIG. 10, los parámetros afines del subbloque TL1002 se predicen usando el bloque vecino B2 (1004) mientras que el subbloque TR 1006 usa el bloque vecino B1 (1008).

[0132] En el diseño actual de predicción de movimiento afín en JEM3.0, si el tamaño de una lista de candidatos de conjunto de MVP afines de un bloque actual es menor que 2, el codificador de vídeo rellena la lista de candidatos de conjunto de MVP afines con uno o más conjuntos de MVP afines compuestos duplicando cada uno de los candidatos de AMVP. Esta divulgación puede usar el término "candidato derivado de AMVP" para referirse a un conjunto de MVP afín compuesto duplicando candidatos de AMVP. Sin embargo, si el vector de movimiento de Lista X del punto de control V0 (FIG. 6A) y el vector de movimiento de Lista X del punto de control V1 son iguales en el modelo de movimiento afín de 4 parámetros o los vectores de movimiento de Lista X de los puntos de control V0, V1 y V2 (FIG. 8A) son iguales en el modelo de movimiento afín de 6 parámetros, los MV calculados para cada subbloque del bloque actual son idénticos. Cuando los MV calculados para cada subbloque del bloque actual son idénticos, los resultados son los mismos que los de la predicción convencional compensada por movimiento. Sin embargo, llegar al mismo resultado de predicción con compensación de movimiento usando la predicción de movimiento afín puede dar como resultado una menor eficacia de codificación que la predicción convencional compensada por movimiento. Por lo tanto, incluir en la lista de candidatos de conjunto de MVP afines un candidato derivado de AMVP que proporcione el mismo resultado que la predicción convencional compensada por movimiento representa una oportunidad perdida de incluir en la lista de candidatos de conjunto de MVP afines un conjunto de MVP afines que puede dar como resultado una mejor eficacia de codificación que la predicción convencional compensada por movimiento.

[0133] Por lo tanto, de acuerdo con una técnica de esta divulgación, un codificador de vídeo agrega un desplazamiento a uno o más MVP en un candidato derivado de AMVP para asegurarse de que no todos los MVP dentro del candidato derivado de AMVP sean idénticos. La inclusión de una diversidad de conjuntos de MVP afines en una lista de candidatos de conjunto de MVP afines puede mejorar la probabilidad de que el uso de uno de los conjuntos de MVP afines en la lista de candidatos de conjunto de MVP afines dé como resultado una mayor eficacia de codificación.

[0134] La FIG. 11A ilustra una lista de candidatos de conjunto de MVP afín ejemplar para un modelo de movimiento afín de 4 parámetros, de acuerdo con una técnica de esta divulgación. La FIG. 11B ilustra una lista de candidatos de conjunto de MVP afín ejemplar para un modelo de movimiento afín de 6 parámetros, de acuerdo con una técnica de esta divulgación. Como se muestra en el ejemplo de la FIG. 11A, en el modelo de movimiento afín de 4 parámetros, el codificador de vídeo agrega y desplaza al MVP para el punto de control V1. Como se muestra en el ejemplo de la FIG. 11B, para un modelo de movimiento afín de 6 parámetros, el codificador de vídeo agrega un desplazamiento al MVP para el punto de control V2. En algunos ejemplos, el desplazamiento es diferente para los candidatos derivados de AMVP en diferentes posiciones en la lista de candidatos de conjunto de MVP afines. Por ejemplo, el codificador de vídeo puede usar +4 como desplazamiento para el primer candidato derivado de AMVP y puede usar -4 como desplazamiento para el segundo candidato derivado de AMVP.

[0135] En el ejemplo de la FIG. 11A, un codificador de vídeo genera una lista de candidatos de Lista X 1100 (es decir, un conjunto de MVP afín) para un bloque actual, donde X es 0 o 1. Para generar la lista de candidatos 1100, el codificador de vídeo comprueba candidatos potenciales. Cada uno de los candidatos potenciales es una combinación de un vector de movimiento de Lista X de un bloque seleccionado de los bloques V_A, V_B y V_C (FIG. 6A) y un vector de movimiento de Lista X de un bloque seleccionado entre V_D y V_E (FIG. 6A). Si ambos bloques del candidato potencial especifican un vector de movimiento de Lista X, el codificador de vídeo incluye al candidato potencial como candidato en la lista de candidatos 1100. El codificador de vídeo deja de agregar candidatos después de que la lista de candidatos 1100 incluye dos candidatos.

[0136] Después de comprobar todos los candidatos potenciales, si todavía hay menos de 2 candidatos en la lista de candidatos 1100, el codificador de vídeo puede agregar un primer candidato derivado de AMVP 1102 a la lista de candidatos 1100. El primer candidato derivado de AMVP 1102 especifica un primer predictor de vector de movimiento derivado de AMVP 1104 y un segundo predictor de vector de movimiento derivado de AMVP 1106. El primer predictor de vector de movimiento derivado de AMVP 1104 es un predictor de vector de movimiento para un primer punto de control del bloque actual. El segundo predictor de vector de movimiento derivado de AMVP 1106 es un predictor de vector de movimiento para un segundo punto de control del bloque actual. El codificador de vídeo deriva el primer predictor de vector de movimiento derivado de AMVP 1104 de la misma manera que usa el codificador de vídeo para derivar un primer candidato de vector de movimiento en AMVP. Esta divulgación describe una técnica para derivar candidatos de vector de movimiento en AMVP anterior. El segundo predictor de vector de movimiento derivado de

AMVP 1106 es igual al primer predictor de vector de movimiento derivado de AMVP 1104 más un primer desplazamiento (es decir, Desplazamiento₀) a al menos uno de un componente horizontal o vertical del primer predictor de vector de movimiento derivado de AMVP 1104.

[0137] Si todavía existen menos de 2 candidatos en la lista de candidatos 1100 después de agregar el primer candidato derivado de AMVP 1102 a la lista de candidatos 1100, el codificador de vídeo agrega un segundo candidato derivado de AMVP 1108 a la lista de candidatos 1100. El segundo candidato de movimiento derivado de AMVP 1108 especifica un tercer predictor de vector de movimiento derivado de AMVP 1110 y un cuarto predictor de vector de movimiento derivado de AMVP 1112. El codificador de vídeo deriva el tercer predictor de vector de movimiento derivado de AMVP 1110 de la misma manera que usa el codificador de vídeo para derivar un segundo candidato de vector de movimiento en AMVP. El cuarto predictor de vector de movimiento derivado de AMVP 1112 es igual al tercer predictor de vector de movimiento derivado de AMVP 1110, excepto porque el codificador de vídeo agrega un segundo desplazamiento (es decir, Desplazamiento₁) a al menos uno de un componente horizontal o vertical del tercer predictor de vector de movimiento derivado de AMVP 1110. Si existen 2 candidatos en la lista de candidatos 1100 después de agregar el primer candidato derivado de AMVP 1102 a la lista de candidatos 1100, el codificador de vídeo no agrega el segundo candidato derivado de AMVP 1108 a la lista de candidatos 1100. Si el bloque actual se predice bidireccionalmente, el codificador de vídeo puede repetir el proceso descrito anteriormente para generar la lista de candidatos 1100 con respecto a los vectores de movimiento de la Lista Y, donde Y es igual a 1-X.

[0138] El ejemplo de la FIG. 11B muestra un proceso similar, excepto que se usa un modelo de movimiento afín de 6 parámetros. Por tanto, para generar la lista de candidatos 1120, el codificador de vídeo comprueba candidatos potenciales. Cada uno de los candidatos potenciales es una combinación del vector de movimiento de Lista X de un bloque seleccionado de los bloques V_A, V_B y V_C (FIG. 8A), un vector de movimiento de Lista X de un bloque seleccionado de entre V_D y V_E (FIG. 8A), y un vector de movimiento de Lista X de un bloque seleccionado de entre V_F y V_G (FIG. 8A). Después de comprobar cada una de las combinaciones, si todavía hay menos de 2 candidatos en la lista de candidatos 1120, el codificador de vídeo puede agregar un primer candidato derivado de AMVP 1122 a la lista de candidatos 1120. El primer candidato derivado de AMVP 1122 especifica un primer predictor de vector de movimiento derivado de AMVP 1124 (denominado AMVP₀ en la FIG. 11B), un segundo predictor de vector de movimiento derivado de AMVP 1126 y un tercer predictor de vector de movimiento derivado de AMVP 1128. El primer predictor de vector de movimiento derivado de AMVP 1124 es un predictor de vector de movimiento para un primer punto de control del bloque actual, el segundo predictor de vector de movimiento derivado de AMVP 1126 es un predictor de vector de movimiento para un segundo punto de control del bloque actual, y el tercer predictor de vector de movimiento derivado de AMVP 1128 es un predictor de vector de movimiento para un tercer punto de control del bloque actual. El codificador de vídeo deriva el primer predictor de vector de movimiento derivado de AMVP 1124 de la misma manera que usa el codificador de vídeo para derivar un primer candidato de vector de movimiento en AMVP. El segundo predictor de vector de movimiento derivado de AMVP 1128 es igual al primer predictor de vector de movimiento derivado de AMVP 1126. El tercer predictor de vector de movimiento derivado de AMVP 1128 es igual al primer predictor de vector de movimiento derivado de AMVP más un primer desplazamiento (es decir, Desplazamiento₀) a al menos uno de un componente horizontal o vertical del primer vector de movimiento derivado de AMVP 1124.

[0139] Si todavía existen menos de 2 candidatos en la lista de candidatos 1120 después de agregar el primer candidato derivado de AMVP 1122 a la lista de candidatos 1120, el codificador de vídeo agrega un segundo candidato derivado de AMVP 1130 a la lista de candidatos 1120. El segundo candidato de movimiento derivado de AMVP 1130 especifica un cuarto predictor de vector de movimiento derivado de AMVP 1132 (denominado AMVP₁ en la FIG. 11B), un quinto predictor de vector de movimiento derivado de AMVP 1134 y un sexto predictor de vector de movimiento derivado de AMVP 1136. El codificador de vídeo deriva el cuarto predictor de vector de movimiento derivado de AMVP 1132 de la misma manera que usa el codificador de vídeo para derivar un segundo candidato de vector de movimiento en AMVP. El quinto predictor de vector de movimiento derivado de AMVP 1134 es igual al cuarto vector de movimiento derivado de AMVP 1132. El sexto predictor de vector de movimiento derivado de AMVP 1136 es igual al tercer predictor de vector de movimiento derivado de AMVP 1132 más un segundo desplazamiento (es decir, Desplazamiento₁) a al menos uno de un componente horizontal o vertical del tercer predictor de vector de movimiento derivado de AMVP 1132. Si existen 2 candidatos en la lista de candidatos 1120 después de agregar el primer candidato derivado de AMVP 1122 a la lista de candidatos 1120, el codificador de vídeo no agrega el segundo candidato derivado de AMVP 1130 a la lista de candidatos 1120. Si el bloque actual se predice bidireccionalmente, el codificador de vídeo puede repetir el proceso descrito anteriormente para generar la lista de candidatos 1120 con respecto a la Lista Y, donde Y es igual a 1-X.

[0140] En algunos ejemplos, la predicción del vector de movimiento de los bloques distintos de los bloques vecinos usados para la derivación de MVP en HEVC se puede agregar a la lista de candidatos. En algunos ejemplos, un codificador de vídeo actualiza un MVP global para movimiento afín instantáneo y el codificador de vídeo usa el MVP global para movimiento afín cuando el tamaño de la lista de candidatos es menor que 2. Por ejemplo, un codificador de vídeo puede construir un modelo de movimiento afín global usando los bloques afines disponibles y puede actualizar el modelo de movimiento afín global siempre que el codificador de vídeo reconstruya un bloque afín. El codificador de vídeo puede usar a continuación este modelo de movimiento afín global para generar un MVP global para los siguientes bloques afines.

[0141] En HEVC y JEM3.0, un indicador en la cabecera de fragmento, `mvd_l1_zero_flag`, indica si la MVD para la segunda lista de imágenes de referencia (por ejemplo, Lista 1) es igual a cero y, por lo tanto, no está señalizada en el flujo de bits para mejorar aún más la eficacia de codificación. En otras palabras, un solo indicador en una cabecera de fragmento de un fragmento puede indicar que todas las MVD de la Lista 1 para todos los bloques del fragmento son iguales a 0. El uso de este indicador puede incrementar la eficacia de la codificación al eliminar la necesidad de señalar por separado MVD de la Lista 1 iguales a 0 para cada bloque codificado de AMVP o AF_INTRA del fragmento.

[0142] Sin embargo, de acuerdo con una técnica de esta divulgación, `mvd_l1_zero_flag` puede ser aplicable a determinados modos de codificación y para otros modos, este indicador se ignora incluso si el indicador indica que la MVD es igual a cero. En un ejemplo, este diseño de MVD cero está deshabilitado para el modo de movimiento afín, pero este diseño de MVD cero todavía se mantiene para el intermodo convencional (modo de AMVP). En otras palabras, incluso si el `mvd_l1_zero_flag` de un fragmento indica que todas las MVD de Lista1 en el fragmento son iguales a 0, el codificador de vídeo 20 todavía puede señalar MVD de Lista 1 para bloques del fragmento que están codificados usando un modo de movimiento afín. Aun siendo capaz de señalar las MVD de Lista 1 para bloques que están codificados usando un modo de movimiento afín a pesar de que `mvd_l1_zero_flag` indica que las MVD de Lista 1 son iguales a 0, el codificador de vídeo 20 puede evitar la señalización de MVD de Lista 1 para bloques que no están codificados usando el modo de movimiento afín sin dejar de ser capaz de señalar MVD de Lista 1 para bloques que están codificados usando el modo de movimiento afín. Esto puede dar como resultado una mayor eficacia de codificación. Esta divulgación describe operaciones ejemplares de acuerdo con esta técnica ejemplar a continuación con referencia a la FIG. 19A y FIG. 19B.

[0143] La FIG. 12 es un diagrama de bloques que ilustra un codificador de vídeo ejemplar 20 que puede implementar las técnicas de esta divulgación. La FIG. 12 se proporciona con propósitos explicativos y no se debería considerar limitante de las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Las técnicas de esta divulgación pueden ser aplicables a diversos estándares o procedimientos de codificación.

[0144] En el ejemplo de la FIG. 12, el codificador de vídeo 20 incluye una unidad de procesamiento de predicción 1200, una memoria de datos de vídeo 1201, una unidad de generación residual 1202, una unidad de procesamiento de transformada 1204, una unidad de cuantificación 1206, una unidad de cuantificación inversa 1208, una unidad de procesamiento de transformada inversa 1210, una unidad de reconstrucción 1212, una unidad de filtro 1214, una memoria intermedia de imágenes descodificadas 1216 y una unidad de codificación por entropía 1218. La unidad de procesamiento de predicción 1200 incluye una unidad de procesamiento de interpredicción 1220 y una unidad de procesamiento de intrapredicción 1222. La unidad de procesamiento de interpredicción 1220 puede incluir una unidad de estimación de movimiento y una unidad de compensación de movimiento (no se muestran).

[0145] La memoria de datos de vídeo 1201 se puede configurar para almacenar datos de vídeo que se van a codificar mediante los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 1201 se pueden obtener, por ejemplo, a partir de la fuente de vídeo 18. La memoria intermedia de imágenes descodificadas 1216 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo mediante el codificador de vídeo 20, por ejemplo, en los modos de intracodificación o intercodificación. La memoria de datos de vídeo 1201 y la memoria intermedia de imágenes descodificadas 1216 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como la memoria dinámica de acceso aleatorio (DRAM), incluyendo la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El mismo dispositivo de memoria u otros dispositivos de memoria separados pueden proporcionar una memoria de datos de vídeo 1201 y una memoria intermedia de imágenes descodificadas 1216. En diversos ejemplos, la memoria de datos de vídeo 1201 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera de chip relativo a esos componentes. La memoria de datos de vídeo 1201 puede ser la misma o parte del medio de almacenamiento 19 de la FIG. 1.

[0146] El codificador de vídeo 20 recibe datos de vídeo. El codificador de vídeo 20 puede codificar cada CTU en un fragmento de una imagen de los datos de vídeo. Cada una de las CTU puede estar asociada a bloques de árbol de codificación (CTB) de luma de igual tamaño y a CTB correspondientes de la imagen. Como parte de la codificación de una CTU, la unidad de procesamiento de predicción 1200 puede realizar una división para dividir los CTB de la CTU en bloques progresivamente más pequeños. Los bloques más pequeños pueden ser bloques de codificación de CU. Por ejemplo, la unidad de procesamiento de predicción 1200 puede dividir un CTB asociado con una CTU de acuerdo con una estructura de árbol.

[0147] El codificador de vídeo 20 puede codificar las CU de una CTU para generar representaciones codificadas de las CU (es decir, CU codificadas). Como parte de la codificación de una CU, la unidad de procesamiento de predicción 1200 puede dividir los bloques de codificación asociados a la CU entre una o más PU de la CU. Por tanto, cada PU puede estar asociada a un bloque de predicción de luma y a bloques de predicción de croma correspondientes. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir PU que tienen diversos tamaños. Como se indica anteriormente, el tamaño de una CU se puede referir al tamaño del bloque de codificación de luma de la CU, y el tamaño de una PU se puede referir al tamaño de un bloque de predicción de luma de la PU. Suponiendo que el

tamaño de una CU particular es de $2N \times 2N$, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden admitir tamaños de PU de $2N \times 2N$ o $N \times N$ para la intrapredicción, y tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, o similares, para la interpredicción. El codificador de vídeo 20 y el decodificador de vídeo 30 también pueden admitir una división asimétrica para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para la interpredicción.

[0148] La unidad de procesamiento de interpredicción 1220 puede generar datos predictivos para una PU. Como parte de la generación de datos predictivos para una PU, la unidad de procesamiento de interpredicción 1220 realiza interpredicción en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos de la PU e información de movimiento para la PU. La unidad de procesamiento de interpredicción 1220 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento I, todas las PU se intrapredicen. Por consiguiente, si la PU está en un fragmento I, la unidad de procesamiento de interpredicción 1220 no realiza interpredicción en la PU. Por tanto, para bloques codificados en el modo I, el bloque predicho se forma usando predicción espacial a partir de bloques vecinos previamente codificados dentro de la misma trama. Si una PU está en un fragmento P, la unidad de procesamiento de interpredicción 1220 puede usar interpredicción unidireccional para generar un bloque predictivo de la PU. Si una PU está en un fragmento B, la unidad de procesamiento de interpredicción 1220 puede usar interpredicción unidireccional o bidireccional para generar un bloque predictivo de la PU.

[0149] La unidad de procesamiento de interpredicción 1220 puede aplicar las técnicas para modelos de movimiento afines como se describe en otra parte de esta divulgación. Por ejemplo, la unidad de procesamiento de interpredicción 1220 puede seleccionar un bloque afín de origen, donde el bloque afín de origen es un bloque codificado por afinidad que es espacialmente vecino a un bloque actual. En este ejemplo, la unidad de procesamiento de interpredicción 1220 puede extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual. Además, en este ejemplo, la unidad de procesamiento de interpredicción 1220 puede insertar, en una lista de candidatos de conjunto de MVP afines, un conjunto de MVP afín que incluye los predictores de vector de movimiento para los puntos de control del bloque actual. En este ejemplo, la unidad de procesamiento de interpredicción 1220 puede seleccionar un conjunto de MVP afín en la lista de candidatos de conjunto de MVP afines. Adicionalmente, en este ejemplo, la unidad de procesamiento de interpredicción 1220 puede señalar, en un flujo de bits, MVD que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y los predictores de vectores de movimiento en el conjunto de MVP afín seleccionado. La unidad de procesamiento de interpredicción 1220 también puede señalar, en el flujo de bits, un índice que indica una posición en la lista de candidatos de conjunto de MVP afines del conjunto de MVP afín seleccionado.

[0150] La unidad de procesamiento de intrapredicción 1222 puede generar datos predictivos para una PU realizando una intrapredicción en la PU. Los datos predictivos para la PU pueden incluir bloques predictivos de la PU y diversos elementos de sintaxis. La unidad de procesamiento de intrapredicción 1222 puede realizar una intrapredicción en las PU en fragmentos I, fragmentos P y fragmentos B.

[0151] Para realizar una intrapredicción en una PU, la unidad de procesamiento de intrapredicción 1222 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos predictivos para la PU. La unidad de procesamiento de intrapredicción 1222 puede usar muestras de bloques de muestras de PU vecinas para generar un bloque predictivo para una PU. Las PU vecinas pueden estar arriba, arriba y a la derecha, arriba y a la izquierda, o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha y de arriba abajo, para las PU, CU y CTU. La unidad de procesamiento de intrapredicción 1222 puede usar diversos números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional. En algunos ejemplos, el número de modos de intrapredicción puede depender del tamaño de la región asociada a la PU.

[0152] La unidad de procesamiento de predicción 1200 puede seleccionar los datos predictivos para las PU de una CU de entre los datos predictivos generados por la unidad de procesamiento de interpredicción 1220 para las PU, o los datos predictivos generados por la unidad de procesamiento de intrapredicción 1222 para las PU. En algunos ejemplos, la unidad de procesamiento de predicción 1200 selecciona los datos predictivos para las PU de la CU en base a unas métricas de velocidad/distorsión de los conjuntos de datos predictivos. Los bloques predictivos de los datos predictivos seleccionados pueden denominarse en el presente documento bloques predictivos seleccionados.

[0153] La unidad de generación residual 1202 puede generar, en base a los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) para una CU y los bloques predictivos seleccionados (por ejemplo, bloques predictivos de luma, Cb y Cr) para las PU de la CU, bloques residuales (por ejemplo, bloques residuales de luma, Cb y Cr) para la CU. Por ejemplo, la unidad de generación residual 1202 puede generar los bloques residuales de la CU de modo que cada muestra en los bloques residuales tenga un valor igual a una diferencia entre una muestra en un bloque de codificación de la CU y una muestra correspondiente en un bloque predictivo seleccionado correspondiente de una PU de la CU.

[0154] La unidad de procesamiento de transformada 1204 puede realizar la división de los bloques residuales de una CU en bloques de transformada de TU de la CU. Por ejemplo, la unidad de procesamiento de transformada 1204 puede realizar partición de árbol cuaternario para dividir los bloques residuales de la CU en bloques de transformada

de TU de la CU. Por tanto, una TU puede estar asociada a un bloque de transformada de luma y a dos bloques de transformada de croma. Los tamaños y las posiciones de los bloques de transformada de luma y croma de las TU de una CU pueden o no estar basados en los tamaños y las posiciones de bloques de predicción de las PU de la CU. Una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados a cada una de las regiones. Las TU de una CU pueden corresponder a nodos hoja del RQT.

[0155] La unidad de procesamiento de transformada 1204 puede generar bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a los bloques de transformada de la TU. La unidad de procesamiento de transformada 1204 puede aplicar diversas transformadas a un bloque de transformada asociado a una TU. Por ejemplo, la unidad de procesamiento de transformada 1204 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar a un bloque de transformada. En algunos ejemplos, la unidad de procesamiento de transformada 1204 no aplica transformadas a un bloque de transformada. En dichos ejemplos, el bloque de transformada se puede tratar como un bloque de coeficientes de transformada.

[0156] La unidad de cuantificación 1206 puede cuantificar los coeficientes de transformada en un bloque de coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de n bits se puede redondear por defecto hasta un coeficiente de transformada de m bits durante la cuantificación, donde n es mayor que m . La unidad de cuantificación 1206 puede cuantificar un bloque de coeficientes asociado a una TU de una CU en base a un valor de parámetro de cuantificación (QP) asociado a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados a una CU, ajustando el valor de QP asociado a la CU. La cuantificación puede introducir pérdida de información. Por tanto, los coeficientes de transformada cuantificados pueden tener una precisión menor que los originales.

[0157] La unidad de cuantificación inversa 1208 y la unidad de procesamiento de transformada inversa 1210 pueden aplicar una cuantificación inversa y transformadas inversas a un bloque de coeficientes, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes. La unidad de reconstrucción 1212 puede añadir el bloque residual reconstruido a las muestras correspondientes de uno o más bloques predictivos generados por la unidad de procesamiento de predicción 1200 para generar un bloque de transformada reconstruido asociado a una TU. Reconstruyendo bloques de transformada para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir los bloques de codificación de la CU.

[0158] La unidad de filtro 1214 puede realizar una o más operaciones de reducción de efecto bloque para reducir los artefactos de efecto bloque en los bloques de codificación asociados a una CU. La memoria intermedia de imágenes descodificadas 1216 puede almacenar los bloques de codificación reconstruidos después de que la unidad de filtro 1214 realice las una o más operaciones de reducción de efecto bloque en los bloques de codificación reconstruidos. La unidad de procesamiento de interpredicción 1220 puede usar una imagen de referencia que contiene los bloques de codificación reconstruidos para realizar una interpredicción en las PU de otras imágenes. Además, la unidad de procesamiento de intrapredicción 1222 puede usar bloques de codificación reconstruidos de la memoria intermedia de imágenes descodificadas 1216 para realizar una intrapredicción en otras PU de la misma imagen que la CU.

[0159] La unidad de codificación por entropía 1218 puede recibir datos desde otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 1218 puede recibir bloques de coeficientes desde la unidad de cuantificación 1206 y puede recibir elementos de sintaxis desde la unidad de procesamiento de predicción 1200. La unidad de codificación por entropía 1218 puede realizar una o más operaciones de codificación por entropía en los datos para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 1218 puede realizar una operación de CABAC, una operación de codificación de longitud variable adaptativa al contexto (CAVLC), una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una operación de codificación por entropía por división de intervalos de probabilidad (PIPE), una operación de codificación exponencial-Golomb u otro tipo de operación de codificación por entropía en los datos. El codificador de vídeo 20 puede proporcionar un flujo de bits que incluye datos codificados por entropía generados por la unidad de codificación por entropía 1218. Por ejemplo, el flujo de bits puede incluir datos que representan valores de coeficientes de transformada para una CU.

[0160] La FIG. 13 es un diagrama de bloques que ilustra un descodificador de vídeo ejemplar 30 que está configurado para implementar las técnicas de esta divulgación. La FIG. 13 se proporciona con propósitos explicativos y no se limita a las técnicas ampliamente ejemplificadas y descritas en la presente divulgación. Con propósitos explicativos, la presente divulgación describe un descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de la presente divulgación pueden ser aplicables a otros estándares o procedimientos de codificación.

[0161] En el ejemplo de la FIG. 13, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 1300, una memoria de datos de vídeo 1301, una unidad de procesamiento de predicción 1302, una unidad de cuantificación inversa 1304, una unidad de procesamiento de transformada inversa 1306, una unidad de

reconstrucción 1308, una unidad de filtro 1310 y una memoria intermedia de imágenes descodificadas 1312. La unidad de procesamiento de predicción 1302 incluye una unidad de compensación de movimiento 1314 y una unidad de procesamiento de intrapredicción 1316. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

[0162] La memoria de datos de vídeo 1301 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado, para ser descodificado por los componentes del descodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 1301 pueden obtenerse, por ejemplo, a partir de un medio legible por ordenador 16, por ejemplo, desde una fuente de vídeo local, tal como una cámara, mediante comunicación de datos de vídeo por red alámbrica o inalámbrica, o accediendo a medios de almacenamiento físico de datos. La memoria de datos de vídeo 1301 puede formar una memoria intermedia de imágenes codificadas (CPB) que almacena datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. La memoria intermedia de imágenes descodificadas 1312 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la descodificación de datos de vídeo mediante el descodificador de vídeo 30, por ejemplo, en los modos de intracodificación o intercodificación o para la salida. La memoria de datos de vídeo 1301 y la memoria intermedia de imágenes descodificadas 1312 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como la memoria dinámica de acceso aleatorio (DRAM), incluyendo la DRAM síncrona (SDRAM), la RAM magnetorresistiva (MRAM), la RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. El mismo dispositivo de memoria u otros dispositivos de memoria separados pueden proporcionar una memoria de datos de vídeo 1301 y una memoria intermedia de imágenes descodificadas 1312. En diversos ejemplos, la memoria de datos de vídeo 1301 puede estar en un chip con otros componentes del descodificador de vídeo 30, o fuera de chip relativo a esos componentes. La memoria de datos de vídeo 1301 puede ser la misma o parte del medio de almacenamiento 28 de la FIG. 1.

[0163] La memoria de datos de vídeo 1301 recibe y almacena datos de vídeo codificados (por ejemplo, unidades de NAL) de un flujo de bits. La unidad de descodificación por entropía 1300 puede recibir datos de vídeo codificados (por ejemplo, unidades de NAL) desde la memoria de datos de vídeo 1301 y puede analizar las unidades de NAL para obtener elementos de sintaxis. La unidad de descodificación por entropía 1300 puede descodificar por entropía elementos de sintaxis codificados por entropía en las unidades de NAL. La unidad de procesamiento de predicción 1302, la unidad de cuantificación inversa 1304, la unidad de procesamiento de transformada inversa 1306, la unidad de reconstrucción 1308 y la unidad de filtro 1310 pueden generar datos de vídeo descodificados en base a los elementos de sintaxis extraídos del flujo de bits. La unidad de descodificación por entropía 1300 puede realizar un proceso en general recíproco al de la unidad de codificación por entropía 1218.

[0164] Además de obtener elementos de sintaxis del flujo de bits, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU. Para realizar la operación de reconstrucción en una CU, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Realizando la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir bloques residuales de la CU.

[0165] Como parte de realizar una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 1304 puede realizar la cuantificación inversa, es decir, descuantificar, los bloques de coeficientes asociados a la TU. Después de que la unidad de cuantificación inversa 1304 haya realizado la cuantificación inversa de un bloque de coeficientes, la unidad de procesamiento de transformada inversa 1306 puede aplicar una o más transformadas inversas al bloque de coeficientes para generar un bloque residual asociado a la TU. Por ejemplo, la unidad de procesamiento de transformada inversa 1306 puede aplicar al bloque de coeficientes una DCT inversa, una transformada entera inversa, una transformada inversa de Karhunen-Loeve (KLT), una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa.

[0166] La unidad de cuantificación inversa 1304 puede realizar técnicas particulares de esta divulgación. Por ejemplo, para al menos un grupo de cuantificación respectivo de una pluralidad de grupos de cuantificación dentro de un CTB de una CTU de una imagen de los datos de vídeo, la unidad de cuantificación inversa 1304 puede derivar, en base al menos en parte a la información de cuantificación local señalizada en el flujo de bits, un parámetro de cuantificación respectivo para el grupo de cuantificación respectivo. Adicionalmente, en este ejemplo, la unidad de cuantificación inversa 1304 puede cuantificar de forma inversa, en base al parámetro de cuantificación respectivo para el grupo de cuantificación respectivo, al menos un coeficiente de transformada de un bloque de transformada de una TU de una CU de la CTU. En este ejemplo, el grupo de cuantificación respectivo se define como un grupo de CU o bloques de codificación sucesivos, en orden de codificación, de modo que los límites del grupo de cuantificación respectivo deben ser los límites de las CU o bloques de codificación y el tamaño del grupo de cuantificación respectivo es mayor o igual a un umbral. El descodificador de vídeo 30 (por ejemplo, unidad de procesamiento de transformada inversa 1306, unidad de reconstrucción 1308 y unidad de filtro 1310) puede reconstruir, en base a coeficientes de transformada cuantificados inversos del bloque de transformada, un bloque de codificación de la CU.

[0167] Si se codifica una PU usando intrapredicción, la unidad de procesamiento de intrapredicción 1316 puede realizar intrapredicción para generar bloques predictivos de la PU. La unidad de procesamiento de intrapredicción 1316 puede usar un modo de intrapredicción para generar los bloques predictivos de la PU en base a bloques de muestras espacialmente vecinas. La unidad de procesamiento de intrapredicción 1316 puede determinar el modo de

intrapredicción para la PU en base a uno o más elementos de sintaxis obtenidos a partir del flujo de bits.

[0168] Si una PU se codifica usando interpredicción, la unidad de compensación de movimiento 1314 puede determinar información de movimiento para la PU. La unidad de compensación de movimiento 1314 puede determinar, en base a la información de movimiento de la PU, uno o más bloques de referencia. La unidad de compensación de movimiento 1314 puede generar, en base a los uno o más bloques de referencia, bloques predictivos (por ejemplo, bloques predictivos de luma, Cb y Cr) para la PU.

[0169] La unidad de compensación de movimiento 1314 puede aplicar las técnicas para modelos de movimiento afines como se describe en otra parte de esta divulgación. Por ejemplo, la unidad de compensación de movimiento 1314 puede seleccionar un bloque afín de origen, donde el bloque afín de origen es un bloque codificado de forma afín que es vecino espacialmente a un bloque actual. En este ejemplo, la unidad de compensación de movimiento 1314 puede extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual. En este ejemplo, la unidad de compensación de movimiento 1314 inserta, en una lista de candidatos de conjunto de MVP afines, un conjunto de MVP afín que incluye los predictores de vector de movimiento para los puntos de control del bloque actual. Además, la unidad de compensación de movimiento 1314 determina, en base a un índice señalado en un flujo de bits, un conjunto de MVP afín seleccionado en la lista de candidatos de conjunto de MVP afines. En este ejemplo, la unidad de descodificación por entropía 1300 puede obtener, del flujo de bits, MVD que indican diferencias entre los vectores de movimiento de los puntos de control del bloque actual y los predictores de vector de movimiento en el conjunto de MVP afín seleccionado. La unidad de compensación de movimiento 1314 puede, en este ejemplo, determinar, en base a los predictores de vector de movimiento incluidos en el conjunto de MVP afín seleccionado y las MVD, vectores de movimiento de los puntos de control del bloque actual. En este ejemplo, la unidad de compensación de movimiento 1314 puede generar, en base a los vectores de movimiento de los puntos de control del bloque actual, un bloque predictivo.

[0170] La unidad de reconstrucción 1308 puede usar bloques de transformada (por ejemplo, bloques de transformada de luma, Cb y Cr) para TU de una CU y los bloques predictivos (por ejemplo, bloques de luma, Cb y Cr) de las PU de la CU, es decir, cualquiera de los datos de intrapredicción o de los datos de interpredicción, según corresponda, para reconstruir los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) para la CU. Por ejemplo, la unidad de reconstrucción 1308 puede agregar muestras de los bloques de transformada (por ejemplo, los bloques de transformada de luma, Cb y Cr) a las muestras correspondientes de los bloques predictivos (por ejemplo, los bloques predictivos de luma, Cb y Cr) para reconstruir los bloques de codificación (por ejemplo, bloques de codificación de luma, Cb y Cr) de la CU.

[0171] La unidad de filtro 1310 puede realizar una operación de reducción de efecto bloque para reducir las distorsiones de efecto bloque asociadas con los bloques de codificación de la CU. El descodificador de vídeo 30 puede almacenar los bloques de codificación de la CU en la memoria intermedia de imágenes descodificadas 1312. La memoria intermedia de imágenes descodificadas 1312 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intrapredicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, en base a los bloques de la memoria intermedia de imágenes descodificadas 1312, operaciones de intrapredicción o de interpredicción para PU de otras CU.

[0172] La FIG. 14A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación. Los diagramas de flujo de la presente divulgación se proporcionan como ejemplos. Otros ejemplos de acuerdo con las técnicas de esta divulgación pueden incluir más, menos o diferentes acciones, o pueden realizar acciones en diferentes órdenes.

[0173] Como se describe anteriormente, de acuerdo con una o más técnicas de esta divulgación, se puede usar un modelo de movimiento afín para una dirección de interpredicción para mejorar la señalización de un modelo de movimiento afín para otra dirección de interpredicción. La FIG. 14A y la FIG. 14B muestran operaciones ejemplares de acuerdo con dichas técnicas.

[0174] En el ejemplo de la FIG. 14A, el codificador de vídeo 20 determina, en base a un primer vector de movimiento de un punto de control de un modelo de movimiento afín de un bloque actual de los datos de vídeo, un segundo vector de movimiento del punto de control del modelo de movimiento afín del bloque actual (1400). El primer vector de movimiento corresponde a la Lista X (donde X es 0 o 1) y el segundo vector de movimiento corresponde a la Lista Y (donde Y es 1-X). En un ejemplo para determinar el segundo vector de movimiento del punto de control, el codificador de vídeo 20 determina primero el vector de movimiento de la Lista X del punto de control. El codificador de vídeo 20 puede determinar el vector de movimiento de la Lista X del punto de control de acuerdo con cualquiera de los ejemplos proporcionados en esta divulgación. Además, el codificador de vídeo 20 puede realizar una búsqueda para determinar un vector de movimiento de Lista Y para el punto de control que proporciona un mejor coste de distorsión de velocidad dado el vector de movimiento de Lista X para el punto de control. En este ejemplo, el codificador de vídeo 20 puede señalar una diferencia de vector de movimiento que indica una diferencia entre el vector de movimiento de Lista X para el punto de control y el vector de movimiento de Lista Y para el punto de control.

[0175] Además, el codificador de vídeo 20 genera, en base al modelo de movimiento afín del bloque actual, un bloque predictivo (1402). El codificador de vídeo 20 puede generar el bloque predictivo de acuerdo con los ejemplos proporcionados en otra parte de esta descripción. Por ejemplo, el codificador de vídeo 20 puede usar el vector de movimiento de Lista X y Lista Y de los puntos de control del modelo de movimiento afín del bloque actual para determinar los vectores de movimiento de Lista X y Lista Y para subbloques del bloque actual, y puede aplicar a continuación filtros de interpolación de compensación de movimiento para generar bloques predictivos para cada uno de los subbloques, generando de este modo el bloque predictivo para el bloque actual.

[0176] En el ejemplo de la FIG. 14A, el codificador de vídeo 20 también genera datos usados para descodificar el bloque actual en base al bloque predictivo (1404). El codificador de vídeo 20 puede generar los datos usados para descodificar el bloque actual de acuerdo con cualquiera de los ejemplos proporcionados en otra parte de esta divulgación. Por ejemplo, el codificador de vídeo 20 puede generar datos residuales, aplicar una transformación a los datos residuales, cuantificar los datos residuales transformados y aplicar codificación por entropía a elementos de sintaxis que representan los datos residuales transformados cuantificados, como se describe en otra parte de esta divulgación.

[0177] La FIG. 14B es un diagrama de flujo que ilustra una operación ejemplar para descodificación de datos de vídeo de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 14B, el descodificador de vídeo 30 determina, en base a un primer vector de movimiento de un punto de control de un modelo de movimiento afín de un bloque actual de los datos de vídeo, un segundo vector de movimiento del punto de control del modelo de movimiento afín del bloque actual (1420). El primer vector de movimiento corresponde a una primera lista de imágenes de referencia (es decir, la Lista X, donde X es 0 o 1). El segundo vector de movimiento corresponde a una segunda lista de imágenes de referencia diferente (es decir, la Lista Y, donde Y es igual a 1-X). En este ejemplo, el descodificador de vídeo 30 puede determinar el vector de movimiento de la Lista X para el punto de control de acuerdo con otros ejemplos proporcionados en esta divulgación. Además, en este ejemplo, para determinar el vector de movimiento de la Lista Y para el punto de control, el descodificador de vídeo 30 puede obtener, a partir del flujo de bits, una diferencia del vector de movimiento que indica una diferencia entre el vector de movimiento de la Lista X para el punto de control y el vector de movimiento de la Lista Y para el punto de control. En este ejemplo, el descodificador de vídeo 30 puede añadir la diferencia del vector de movimiento al vector de movimiento de Lista X del punto de control para determinar el vector de movimiento de Lista Y para el punto de control.

[0178] Adicionalmente, el descodificador de vídeo 30 genera, en base al modelo de movimiento afín del bloque actual, un bloque predictivo (1422). El descodificador de vídeo 30 puede generar el bloque predictivo de la misma manera que el codificador de vídeo 20 en la FIG. 14A. El descodificador de vídeo 30 puede reconstruir el bloque actual en base a datos residuales y el bloque predictivo (1424). Por ejemplo, el descodificador de vídeo 30 puede reconstruir el bloque actual al menos en parte añadiendo muestras de los datos residuales a las muestras correspondientes del bloque predictivo.

[0179] La FIG. 15A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación. Como se analiza anteriormente, de acuerdo con algunas técnicas de esta divulgación, los MV de puntos de control de un bloque actual se pueden usar como el candidato de MVP para los MV de otros puntos de control del bloque actual. La FIG. 15A y la FIG. 15B muestran operaciones ejemplares de acuerdo con estas técnicas.

[0180] En el ejemplo de la FIG. 15A, el codificador de vídeo 20 determina, en base a un vector de movimiento de un primer punto de control de un modelo de movimiento afín de un bloque actual de los datos de vídeo, un vector de movimiento de un segundo punto de control del modelo de movimiento afín del bloque actual (1500). Por ejemplo, el codificador de vídeo 20 puede incluir el vector de movimiento del primer punto de control (por ejemplo, un punto de control superior izquierdo) del modelo de movimiento afín del bloque actual como candidato en una lista de candidatos usada para la predicción del vector de movimiento del segundo punto de control (por ejemplo, un punto de control superior derecho) del modelo de movimiento afín del bloque actual. Otros candidatos en la lista de candidatos pueden incluir vectores de movimiento de los puntos de control correspondientes (por ejemplo, puntos de control de la parte superior izquierda) de los bloques codificados por afinidad. En este ejemplo, el codificador de vídeo 20 puede seleccionar a continuación un candidato de la lista de candidatos (por ejemplo, en base al coste de distorsión de la velocidad). Además, en este ejemplo, el codificador de vídeo 20 puede usar a continuación el vector de movimiento del candidato seleccionado como predictor del vector de movimiento para el segundo punto de control. En algunos ejemplos, el codificador de vídeo 20 señaliza una MVD que indica una diferencia entre el vector de movimiento del candidato seleccionado y el predictor de vector de movimiento para el segundo punto de control.

[0181] Además, el codificador de vídeo 20 genera, en base al modelo de movimiento afín del bloque actual, un bloque predictivo (1502). Adicionalmente, el codificador de vídeo 20 genera datos usados para descodificar el bloque actual en base al bloque predictivo (1504). El codificador de vídeo 20 puede generar el bloque predictivo y generar los datos usados para descodificar el bloque actual de acuerdo con ejemplos proporcionados en otra parte de esta divulgación.

[0182] La FIG. 15B es un diagrama de flujo que ilustra una operación ejemplar para descodificación de datos de vídeo de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 15B, el descodificador de vídeo 30 determina, en base a un vector de movimiento de un primer punto de control de un modelo de movimiento afín de un bloque actual de los datos de vídeo, un vector de movimiento de un segundo punto de control del modelo de movimiento afín del bloque actual (1520). El descodificador de vídeo 30 puede determinar el vector de movimiento del segundo punto de control del modelo de movimiento afín del bloque actual de la misma manera que se describe anteriormente con respecto al codificador de vídeo 20 en la FIG. 15A. Para determinar el vector de movimiento del segundo punto de control, el descodificador de vídeo 30 puede obtener de un flujo de bits un índice que indica un candidato seleccionado en una lista de candidatos. La lista de candidatos puede incluir vectores de movimiento de puntos de control, incluyendo un vector de movimiento del primer punto de control del bloque actual. En algunos ejemplos, el descodificador de vídeo 30 puede determinar el vector de movimiento del segundo punto de control añadiendo una MVD señalizada al vector de movimiento del candidato seleccionado.

[0183] Adicionalmente, el descodificador de vídeo 30 genera, en base al modelo de movimiento afín del bloque actual, un bloque predictivo (1522). El descodificador de vídeo 30 reconstruye el bloque en base al bloque predictivo (1524). El descodificador de vídeo 30 puede generar el bloque predictivo y reconstruir el bloque de acuerdo con ejemplos proporcionados en otra parte de esta divulgación. Por ejemplo, el descodificador de vídeo 30 puede reconstruir el bloque en base al bloque predictivo y los datos residuales descodificados.

[0184] La FIG. 16A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación. Como se analiza anteriormente, de acuerdo con algunas técnicas de esta divulgación, un codificador de vídeo puede usar modelos de movimiento de uno o más bloques afines vecinos de un bloque actual para predecir un modelo de movimiento afín actual. En el ejemplo de la FIG. 16A, el codificador de vídeo 20 usa un modelo de movimiento de un único bloque afín vecino para predecir al menos dos puntos de control de un modelo de movimiento afín para un bloque actual (1600). Además, el codificador de vídeo 20 genera, en base al modelo de movimiento afín del bloque actual, un bloque predictivo (1602). El codificador de vídeo 20 genera a continuación datos usados para descodificar el bloque actual en base al bloque predictivo (1604). El codificador de vídeo 20 puede generar los datos de acuerdo con ejemplos proporcionados en otra parte de esta divulgación.

[0185] La FIG. 16B es un diagrama de flujo que ilustra una operación ejemplar para descodificación de datos de vídeo de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 16B, el descodificador de vídeo 30 usa un modelo de movimiento de un único bloque afín vecino para predecir un modelo de movimiento afín para un bloque actual (1620). Además, el descodificador de vídeo 30 genera, en base al modelo de movimiento afín del bloque actual, un bloque predictivo (1622). El descodificador de vídeo 30 puede reconstruir a continuación el bloque actual en base al bloque predictivo (1624). El descodificador de vídeo 30 puede generar el bloque predictivo y reconstruir el bloque actual de acuerdo con cualquiera de los ejemplos proporcionados en otra parte de esta divulgación.

[0186] La FIG. 17 es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación. La FIG. 17 y la FIG. 18 son diagramas de flujo más detallados para las operaciones enumeradas en la FIG. 16A y la FIG. 16B. En el ejemplo de la FIG. 17, el codificador de vídeo 20 selecciona un bloque afín de origen (1700). El bloque afín de origen es un bloque codificado por afinidad que es vecino espacialmente a un bloque actual. El codificador de vídeo 20 puede seleccionar el bloque afín de origen de varias formas. Por ejemplo, el codificador de vídeo 20 puede determinar que el bloque afín de origen es un bloque codificado por afinidad que aparece primero de la pluralidad de bloques vecinos visitados en un orden de visita predefinido. En algunos ejemplos, el codificador de vídeo 20 puede determinar que el bloque afín de origen es un bloque codificado por afinidad disponible que aparece primero de la pluralidad de bloques vecinos de acuerdo con una pluralidad de conjuntos de prioridad predefinidos en base a un orden de visita predefinido. Un bloque codificado por afinidad no se considera disponible si el bloque codificado por afinidad no está en uno de los conjuntos de prioridad predefinidos. En otra parte de esta divulgación se describen diversos ejemplos de conjuntos de prioridad.

[0187] Adicionalmente, el codificador de vídeo 20 puede extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual (1702). Por ejemplo, el codificador de vídeo 20 puede construir un modelo de movimiento afín definido por los parámetros afines determinados por los vectores de movimiento de los puntos de control del bloque de origen afín. El codificador de vídeo 20 puede derivar a continuación los vectores de movimiento de los puntos de control del bloque actual (o los llamados extrapolados) usando el modelo de movimiento afín construido. Por ejemplo, para extrapolar un vector de movimiento de un punto de control del bloque actual, el codificador de vídeo 20 puede usar los vectores de movimiento del modelo de movimiento afín construido y la posición (x, y) del punto de control del bloque actual en la ecuación 2 para determinar un vector de movimiento del punto de control.

[0188] Además, el codificador de vídeo 20 puede insertar, en una lista de candidatos de conjunto de MVP afines, un conjunto de MVP afín que incluye los predictores de vector de movimiento para los puntos de control del bloque actual (1704). En algunos ejemplos, el codificador de vídeo 20 también puede incluir un conjunto de MVP afín convencional en el conjunto de MVP afín. Por ejemplo, en un ejemplo, el codificador de vídeo 20 puede determinar un primer predictor de vector de movimiento como un vector de movimiento de un bloque adyacente al primer punto de control del bloque actual. En este ejemplo, el codificador de vídeo 20 determina un segundo predictor de vector de movimiento

como un vector de movimiento de un bloque adyacente al segundo punto de control del bloque actual (por ejemplo, bloques A, B o C; o bloques D o E de la FIG. 6A). En este ejemplo, el codificador de vídeo 20 inserta, en la lista de candidatos de conjunto de MVP afines, un conjunto de MVP afín que incluye el primer predictor de vector de movimiento y el segundo predictor de vector de movimiento.

[0189] En algunos ejemplos, el codificador de vídeo 20 selecciona un segundo bloque afín de origen. El segundo bloque afín de origen es un bloque codificado por afinidad diferente que es vecino espacialmente al bloque actual. En este ejemplo, el codificador de vídeo 20 extrapola los vectores de movimiento de los puntos de control del segundo bloque afín de origen para determinar segundos predictores de vectores de movimiento para los puntos de control del bloque actual. Además, el codificador de vídeo 20 inserta un segundo conjunto de MVP afín en la lista de candidatos de conjunto de MVP afines. El segundo conjunto de MVP afín incluye los segundos predictores de vector de movimiento para los puntos de control del bloque actual.

[0190] Posteriormente, el codificador de vídeo 20 selecciona un conjunto de MVP afín en la lista de candidatos de conjunto de MVP afines (1706). El codificador de vídeo 20 puede seleccionar el conjunto de MVP afín en base a un análisis de distorsión de la velocidad de los conjuntos de MVP afines en la lista de candidatos de conjunto de MVP afines.

[0191] El codificador de vídeo 20 puede señalar, en un flujo de bits, MVD que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y los predictores de vectores de movimiento en el conjunto de MVP afín seleccionado (1708). Además, el codificador de vídeo 20 puede señalar, en el flujo de bits, un índice que indica una posición en la lista de candidatos de conjunto de MVP afines del conjunto de MVP afín seleccionado (1710).

[0192] La FIG. 18 es un diagrama de flujo que ilustra una operación ejemplar para descodificación de datos de vídeo de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 18, el descodificador de vídeo 30 selecciona un bloque afín de origen (1800). El bloque afín de origen es un bloque codificado por afinidad que es vecino espacialmente a un bloque actual. El descodificador de vídeo 30 puede seleccionar el bloque afín de origen de la misma manera que el codificador de vídeo 20, como se describe en otra parte de esta divulgación.

[0193] Adicionalmente, el descodificador de vídeo 30 extrapola vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual (1802). El descodificador de vídeo 30 inserta, en una lista de candidatos de conjunto de MVP afines, un conjunto de MVP afín que incluye los predictores de vector de movimiento para los puntos de control del bloque actual (1804). El descodificador de vídeo 30 puede extrapolar los vectores de movimiento de los puntos de control e insertar el conjunto de MVP afín de la misma manera que el codificador de vídeo 20, como se describe en otra parte de esta divulgación. El descodificador de vídeo 30 también puede añadir los conjuntos de MVP afines adicionales a la lista de candidatos de conjunto de MVP afines como se describe anteriormente con respecto al codificador de vídeo 20.

[0194] Además, el descodificador de vídeo 30 determina, en base a un índice señalado en un flujo de bits, un conjunto de MVP afín seleccionado en la lista de candidatos de conjunto de MVP afines (1806). El descodificador de vídeo 30 obtiene, de un flujo de bits, MVD que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y los predictores de vectores de movimiento en el conjunto de MVP afín seleccionado (1808). Además, el descodificador de vídeo 30 determina, en base a los predictores de vector de movimiento incluidos en el conjunto de MVP afín seleccionado y las MVD, los vectores de movimiento de los puntos de control del bloque actual (1810). Por ejemplo, el descodificador de vídeo 30 puede añadir las MVD a los predictores de vector de movimiento correspondientes para determinar los vectores de movimiento de los puntos de control del bloque actual.

[0195] El descodificador de vídeo 30 puede generar a continuación, en base a los vectores de movimiento de los puntos de control del bloque actual, un bloque predictivo (1812). El descodificador de vídeo 30 puede reconstruir el bloque actual en base a datos residuales y el bloque predictivo (1814). El descodificador de vídeo 30 puede generar el bloque predictivo y reconstruir el bloque actual de acuerdo con ejemplos proporcionados en otra parte de esta divulgación.

[0196] La FIG. 19A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación. Como se analiza anteriormente, de acuerdo con algunas técnicas de esta divulgación, en el intermodo afín o en el modo de fusión afín, los parámetros afines de cada subbloque (por ejemplo, bloque de 4x4) de un bloque actual se pueden predecir o heredar directamente de los propios bloques vecinos del subbloque. La FIG. 19A y la FIG. 19B muestran operaciones ejemplares de acuerdo con dichas técnicas.

[0197] En el ejemplo de la FIG. 19A, un bloque actual de datos de vídeo se divide en una pluralidad de subbloques. Para cada subbloque respectivo de la pluralidad de subbloques, el codificador de vídeo 20 usa movimiento extrapolado de un respectivo bloque afín vecino para predecir el movimiento afín para el subbloque respectivo (1900). El codificador de vídeo 20 puede extrapolar el movimiento del respectivo bloque afín vecino de la misma forma que se describe en otra parte de esta divulgación. En algunos ejemplos, para usar el movimiento extrapolado del respectivo bloque afín vecino para predecir el movimiento afín para el subbloque respectivo, el codificador de vídeo 20 predice el movimiento

afín para el subbloque respectivo a partir del movimiento extrapolado del bloque afín vecino. En algunos casos, el bloque afín vecino es otro subbloque del bloque actual. En algunos ejemplos, para usar el movimiento extrapolado del respectivo bloque afín vecino, el codificador de vídeo 20 hereda el movimiento afín para cada subbloque respectivo directamente del movimiento extrapolado del bloque afín vecino. En otras palabras, el codificador de vídeo 20 establece los índices de referencia y los vectores de movimiento de los puntos de control del subbloque respectivo iguales a los índices de movimiento y los vectores de movimiento extrapolados de los puntos de control del respectivo bloque afín vecino. Por ejemplo, como se muestra en la FIG. 10, el subbloque 1002 puede heredar los índices de referencia y los vectores de movimiento extrapolados de los puntos de control del bloque afín 1004.

[0198] Además, el codificador de vídeo 20 genera, en base al movimiento afín para los subbloques, un bloque predictivo (1902). Por ejemplo, para cada subbloque respectivo de la pluralidad de subbloques, el codificador de vídeo 20 puede usar el movimiento afín para el subbloque respectivo para generar un subbloque predictivo respectivo para el subbloque respectivo. En este ejemplo, el codificador de vídeo 20 puede usar el movimiento afín para el subbloque respectivo para generar el subbloque predictivo respectivo para el subbloque respectivo de la misma manera descrita en otra parte de esta divulgación para usar el movimiento afín para generar un bloque predictivo. Por ejemplo, el codificador de vídeo 20 puede usar la ecuación (2) para calcular el componente x y el componente y de un vector de movimiento para el subbloque respectivo. El codificador de vídeo 20 puede usar a continuación el vector de movimiento para el subbloque respectivo para determinar un bloque predictivo preliminar o final para el subbloque respectivo. Además, en este ejemplo, el codificador de vídeo 20 puede combinar los subbloques predictivos para generar el bloque predictivo para el bloque actual.

[0199] El codificador de vídeo 20 genera datos usados para descodificar el bloque actual en base al bloque predictivo (1904). El codificador de vídeo 20 puede generar los datos usados para descodificar el bloque actual de acuerdo con cualquiera de los ejemplos correspondientes proporcionados en otra parte de esta divulgación.

[0200] La FIG. 19B es un diagrama de flujo que ilustra una operación ejemplar para descodificación de datos de vídeo de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 16B, un bloque actual de datos de vídeo se divide en una pluralidad de subbloques. Para cada subbloque respectivo de la pluralidad de subbloques, el descodificador de vídeo 30 usa movimiento extrapolado de un respectivo bloque afín vecino para predecir el movimiento afín para el subbloque respectivo (1920). El descodificador de vídeo 30 puede predecir el movimiento afín para el subbloque respectivo de la misma manera que se describe anteriormente con respecto al codificador de vídeo 20. En algunos ejemplos, para usar el movimiento extrapolado del respectivo bloque afín vecino, el descodificador de vídeo 30 predice el movimiento afín para cada subbloque respectivo del movimiento extrapolado del bloque afín vecino. En algunos ejemplos, para usar el movimiento extrapolado del respectivo bloque afín vecino, el descodificador de vídeo 30 hereda el movimiento afín para cada subbloque respectivo directamente del movimiento extrapolado del bloque afín vecino.

[0201] Además, el descodificador de vídeo 30 genera, en base al movimiento afín de los subbloques, un bloque predictivo (1922). El descodificador de vídeo 30 puede generar el bloque predictivo de la misma manera que se describe anteriormente con respecto al codificador de vídeo 20. El descodificador de vídeo 30 reconstruye el bloque en base al bloque predictivo (1924). Por ejemplo, el descodificador de vídeo 30 puede reconstruir el bloque actual añadiendo muestras del bloque predictivo a muestras residuales correspondientes descodificadas del flujo de bits.

[0202] La FIG. 20A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación. Como se analiza anteriormente, de acuerdo con algunas técnicas de esta divulgación, se pueden añadir desplazamientos a los MVP para asegurarse de que no todos los MVP dentro de un conjunto de candidatos de MVP sean idénticos. La FIG. 20A y la FIG. 20B muestran operaciones ejemplares de acuerdo con estas técnicas.

[0203] En particular, en el ejemplo de la FIG. 20A, un primer vector de movimiento es un vector de movimiento de un primer punto de control de un modelo de movimiento afín de un bloque actual. Un segundo vector de movimiento es un vector de movimiento de un segundo punto de control del modelo de movimiento afín del bloque actual. En base a que un número total de candidatos en una lista de candidatos de vectores de movimiento es menor que 2 y el primer vector de movimiento y el segundo vector de movimiento son iguales, el codificador de vídeo 20 añade un desplazamiento a un predictor de vector de movimiento (2000). Como se describe en otra parte de esta divulgación, el desplazamiento puede ser diferente para diferentes posiciones en la lista de candidatos.

[0204] Además, el codificador de vídeo 20 incluye el predictor de vector de movimiento en la lista de candidatos (2002). Por ejemplo, el codificador de vídeo 20 puede incluir el predictor de vector de movimiento en una matriz de predictores de vector de movimiento. Además, el codificador de vídeo 20 selecciona un candidato en la lista de candidatos (2004). El codificador de vídeo 20 puede seleccionar el candidato de modo que el candidato seleccionado dé como resultado el mejor valor de distorsión de velocidad entre los candidatos de la lista de candidatos. Además, el codificador de vídeo 20 usa el candidato seleccionado para determinar un bloque predictivo (2006). Por ejemplo, el codificador de vídeo 20 puede usar los vectores de movimiento especificados por el candidato seleccionado para identificar localizaciones en una imagen de referencia. En este ejemplo, el codificador de vídeo 20 puede determinar el bloque predictivo aplicando una rotación a una copia de un bloque de muestras en las localizaciones identificadas

en la imagen de referencia. En algunos ejemplos, el codificador de vídeo 20 puede usar el candidato seleccionado para determinar un primer bloque predictivo preliminar y también determinar un segundo bloque predictivo (por ejemplo, en base a muestras en una imagen de referencia en una lista de imágenes de referencia diferente). En este ejemplo, el codificador de vídeo 20 puede determinar muestras en el bloque predictivo como promedios ponderados de muestras correspondientes en el primer bloque predictivo preliminar y el segundo bloque predictivo preliminar.

[0205] El codificador de vídeo 20 puede generar a continuación datos residuales en base a muestras del bloque actual y el bloque predictivo (2008). Por ejemplo, el codificador de vídeo 20 puede generar los datos residuales de modo que cada muestra de los datos residuales indique una diferencia entre muestras correspondientes en el bloque actual y el bloque predictivo. Adicionalmente, el codificador de vídeo 20 incluye, en un flujo de bits que comprende una representación codificada de los datos de vídeo, una indicación de un candidato seleccionado en la lista de candidatos (2010).

[0206] La FIG. 20B es un diagrama de flujo que ilustra una operación ejemplar para descodificación de datos de vídeo de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 18B, en base a que un número total de candidatos en una lista de candidatos de vectores de movimiento es menor que 2 y un primer vector de movimiento y un segundo vector de movimiento son iguales, el descodificador de vídeo 30 añade un desplazamiento a un predictor de vector de movimiento (2020). En este ejemplo, el primer vector de movimiento es un vector de movimiento de un primer punto de control de un modelo de movimiento afín de un bloque actual de datos de vídeo. El segundo vector de movimiento es un vector de movimiento de un segundo punto de control del modelo de movimiento afín del bloque actual.

[0207] Además, en el ejemplo de la FIG. 20B, el descodificador de vídeo 30 incluye el predictor de vector de movimiento en la lista de candidatos (2022). El descodificador de vídeo 30 puede determinar a continuación un candidato seleccionado en la lista de candidatos (2024). Adicionalmente, el descodificador de vídeo 30 puede usar el candidato seleccionado para determinar un bloque predictivo (2026). El descodificador de vídeo 30 puede reconstruir a continuación el bloque actual en base al bloque predictivo (2028). En algunos ejemplos, el descodificador de vídeo 30 puede reconstruir el bloque actual en base al bloque predictivo y los datos residuales.

[0208] La FIG. 21A es un diagrama de flujo que ilustra una operación ejemplar para codificación de datos de vídeo de acuerdo con una técnica de esta divulgación. Como se indica anteriormente, de acuerdo con una técnica de esta divulgación, el `mvd_11_zero_flag` puede ser aplicable a determinados modos de codificación y para otros modos, este indicador se ignora incluso si el `mvd_11_zero_flag` indica que la MVD es igual a cero. La FIG. 21A y la FIG. 21B muestran operaciones ejemplares de acuerdo con esta técnica.

[0209] Específicamente, en el ejemplo de la FIG. 21A, el codificador de vídeo 20 incluye, en un flujo de bits, un indicador (por ejemplo, `mvd_11_zero_flag`) que indica si las diferencias de vector de movimiento para las segundas listas de imágenes de referencia (por ejemplo, lista 1) están señalizadas en el flujo de bits (2100). En base al movimiento de un bloque actual de los datos de vídeo que es un modo de movimiento afín, independientemente del valor del indicador, el codificador de vídeo 20 incluye en el flujo de bits una MVD (2102). Por ejemplo, el codificador de vídeo 20 puede incluir en el flujo de bits un elemento de sintaxis que indica un componente vertical de la MVD y un segundo elemento que indica un componente horizontal de la MVD.

[0210] Además, el codificador de vídeo 20 genera, en base al modelo de movimiento afín del bloque actual, un bloque predictivo (2104). El codificador de vídeo 20 puede generar el bloque predictivo en base al modelo de movimiento afín del bloque actual de acuerdo con cualquiera de los ejemplos proporcionados en otra parte de esta divulgación. El codificador de vídeo 20 puede generar a continuación datos usados para descodificar el bloque actual en base al bloque predictivo (2106). Por ejemplo, el codificador de vídeo 20 puede generar datos residuales, aplicar una transformación a los datos residuales, cuantificar los datos residuales transformados y aplicar codificación por entropía a elementos de sintaxis que representan los datos residuales transformados cuantificados, como se describe en otra parte de esta divulgación.

[0211] La FIG. 21B es un diagrama de flujo que ilustra una operación ejemplar para descodificación de datos de vídeo de acuerdo con una técnica de esta divulgación. En el ejemplo de la FIG. 21B, el descodificador de vídeo 30 obtiene, a partir de un flujo de bits, un indicador (por ejemplo, `mvd_11_zero_flag`) que indica si las diferencias de vector de movimiento para las segundas listas de imágenes de referencia (por ejemplo, lista 1) están señalizadas en el flujo de bits (2120).

[0212] Además, en el ejemplo de la FIG. 21B, en base al movimiento de un bloque actual de los datos de vídeo que es un modo de movimiento afín, independientemente del valor del indicador, el descodificador de vídeo 30 obtiene del flujo de bits una MVD (2122). En otras palabras, el descodificador de vídeo 30 descodifica la MVD del flujo de bits. En algunos ejemplos, para obtener la MVD del flujo de bits, el descodificador de vídeo 30 obtiene del flujo de bits un primer elemento de sintaxis que indica un componente vertical de la MVD y un segundo elemento de sintaxis que indica un componente horizontal de la MVD. El descodificador de vídeo 30 determina, en base a la diferencia del vector de movimiento, un modelo de movimiento afín para el bloque actual (2124). Adicionalmente, el descodificador de vídeo 30 genera, en base al modelo de movimiento afín del bloque actual, un bloque predictivo (2126). Además, el

descodificador de vídeo 30 reconstruye el bloque en base al bloque predictivo (2128). El descodificador de vídeo 30 puede generar el bloque predictivo y reconstruir el bloque de acuerdo con cualquiera de los ejemplos proporcionados en otra parte de esta divulgación.

5 **[0213]** Determinados aspectos de esta divulgación se han descrito con respecto a las extensiones del estándar de HEVC con propósitos ilustrativos. Sin embargo, las técnicas descritas en esta divulgación pueden ser útiles para otros procesos de codificación de vídeo, que incluyen otros procesos de codificación de vídeo, estándar o de propiedad, actualmente en desarrollo o aún no desarrollados.

10 **[0214]** Un codificador de vídeo, como se describe en esta divulgación, se puede referir a un codificador de vídeo o a un descodificador de vídeo. De forma similar, una unidad de codificación de vídeo se puede referir a un codificador de vídeo o a un descodificador de vídeo. Asimismo, la codificación de vídeo se puede referir a una codificación de vídeo o a una descodificación de vídeo, según corresponda. En esta divulgación, la frase "basado en" puede indicar
 15 "unidad de vídeo" o "bloque de vídeo" o "bloque" para referirse a uno o más bloques de muestras y estructuras sintácticas usadas para codificar muestras de los uno o más bloques de muestras. Los tipos de unidades de vídeo ejemplares pueden incluir CTU, CU, PU, unidades de transformada (TU), macrobloques, particiones de macrobloque, y así sucesivamente. En algunos contextos, el análisis de las PU puede intercambiarse con el análisis de macrobloques o de particiones de macrobloques. Los tipos de bloques de vídeo ejemplares pueden incluir bloques de árbol de
 20 codificación, bloques de codificación y otros tipos de bloques de datos de vídeo.

[0215] Se debe reconocer que, dependiendo del ejemplo, determinadas acciones o eventos de cualquiera de las técnicas descritas en el presente documento se pueden realizar en una secuencia distinta, se pueden añadir, fusionar o excluir por completo (por ejemplo, no todas las acciones o eventos descritos son necesarios para la puesta en
 25 práctica de las técnicas). Además, en determinados ejemplos, las acciones o eventos se pueden realizar simultáneamente, por ejemplo, a través de procesamientos de múltiples subprocesos, procesamientos de interrupciones o múltiples procesadores, en lugar de secuencialmente.

[0216] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o
 30 en cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en o transmitir a través de un medio legible por ordenador como una o más instrucciones o código, y ejecutarse por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador que correspondan a un medio tangible, tales como medios de almacenamiento de datos, o medios de comunicación que incluyan cualquier medio que facilite la transferencia de un programa
 35 informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o a (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más circuitos de procesamiento para recuperar instrucciones, código y/o estructuras de
 40 datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0217] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco
 45 magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda usar para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica,
 50 el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se debe entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se refieren a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, donde unos discos reproducen normalmente los datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior también se deben incluir dentro del alcance de los medios legibles por ordenador.

[0218] La funcionalidad descrita en esta divulgación se puede realizar mediante una función fija y/o circuitería de
 60 procesamiento programable. Por ejemplo, las instrucciones se pueden ejecutar mediante una función fija y/o circuitería de procesamiento programable. Dicha circuitería de procesamiento puede incluir uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables *in situ* (FPGA) u otra circuitería lógica discreta o
 65 integrada equivalente. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las

técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o de software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos. Los circuitos de procesamiento se pueden acoplar a otros componentes de diversas maneras. Por ejemplo, un circuito de procesamiento puede estar acoplado a otros componentes por medio de una interconexión interna del dispositivo, una conexión de red alámbrica o inalámbrica u otro medio de comunicación.

[0219] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluya uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuados.

REIVINDICACIONES

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

- 5 seleccionar un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es vecino espacialmente a un bloque actual;
- 10 extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual en base a los vectores de movimiento extrapolados;
- 15 insertar, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afín que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual;
- determinar, en base a un índice señalado en un flujo de bits, un conjunto de MVP afín seleccionado en la lista de candidatos de conjunto de MVP afines;
- 20 obtener, de un flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y los predictores de vectores de movimiento en el conjunto de MVP afín seleccionado;
- 25 determinar, en base a los predictores de vector de movimiento incluidos en el conjunto de MVP afín seleccionado y las MVD, los vectores de movimiento de los puntos de control del bloque actual;
- generar, en base a los vectores de movimiento de los puntos de control del bloque actual, un bloque predictivo; y
- 30 reconstruir el bloque actual basado en datos residuales y el bloque predictivo.

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

- 35 seleccionar un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es vecino espacialmente a un bloque actual;
- 40 extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual en base a los vectores de movimiento extrapolados
- insertar, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afín que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual;
- 45 seleccionar un conjunto de MVP afín en la lista de candidatos de conjunto de MVP afines;
- señalar, en un flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y los predictores de vectores de movimiento en el conjunto de MVP afín seleccionado; y
- 50 señalar, en el flujo de bits, un índice que indica una posición en la lista de candidatos de conjunto de MVP afines del conjunto de MVP afín seleccionado.

3. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que los puntos de control del bloque actual incluyen un primer punto de control y un segundo punto de control, comprendiendo además el procedimiento:

- 55 determinar un primer predictor de vector de movimiento como un vector de movimiento de un bloque adyacente al primer punto de control del bloque actual;
- 60 determinar un segundo predictor de vector de movimiento como un vector de movimiento de un bloque adyacente al segundo punto de control del bloque actual; e
- insertar, en la lista de candidatos de conjunto de MVP afines, un conjunto de MVP afín que incluye el primer predictor de vector de movimiento y el segundo predictor de vector de movimiento.

4. El procedimiento de la reivindicación 1 o la reivindicación 2, en el que el bloque actual es adyacente a una pluralidad de bloques vecinos, y la selección del bloque afín de origen comprende:

determinar que el bloque afín de origen es un bloque codificado por afinidad que aparece primero de la pluralidad de bloques vecinos visitados en un orden de visita predefinido.

- 5 **5.** El procedimiento de la reivindicación 1 o la reivindicación 2, en el que el bloque actual es adyacente a una pluralidad de bloques vecinos, y la selección del bloque afín de origen comprende:

10 determinar que el bloque afín de origen es un bloque codificado por afinidad disponible que aparece primero de la pluralidad de bloques vecinos de acuerdo con una pluralidad de conjuntos de prioridad predefinidos en base a un orden de visita predefinido, en el que el bloque afín de origen no se considera disponible si el bloque afín de origen no está en uno de los conjuntos de prioridad predefinidos.

6. El procedimiento de la reivindicación 5, en el que:

15 la pluralidad de conjuntos de prioridad predefinidos incluye un primer conjunto de prioridad y un segundo conjunto de prioridad, y

20 el primer conjunto de prioridad y el segundo conjunto de prioridad se definen de modo que, para cada bloque vecino respectivo de la pluralidad de bloques vecinos:

el bloque vecino respectivo está en el primer conjunto de prioridad si el bloque vecino respectivo tiene una imagen de referencia de Lista X que es igual a una imagen de referencia de Lista X del bloque actual, siendo X 0 o 1; y

25 el bloque vecino respectivo está en el segundo conjunto de prioridad si el bloque vecino respectivo tiene una imagen de referencia de Lista Y que es igual a la imagen de referencia de Lista X del bloque actual, siendo Y igual a 1-X.

7. El procedimiento de la reivindicación 5, en el que:

30 la pluralidad de conjuntos de prioridad predefinidos incluye un primer conjunto de prioridad y un segundo conjunto de prioridad, y

35 el primer conjunto de prioridad y el segundo conjunto de prioridad se definen de modo que, para cada bloque vecino respectivo de la pluralidad de bloques vecinos:

el bloque vecino respectivo está en el primer conjunto de prioridad si el bloque vecino respectivo tiene una imagen de referencia de Lista X distinta a una imagen de referencia de Lista X del bloque actual, siendo X 0 o 1; y

40 el bloque vecino respectivo está en el segundo conjunto de prioridad si el bloque vecino respectivo tiene una imagen de referencia de Lista Y que es distinta a la imagen de referencia de Lista X del bloque actual, siendo Y igual a 1-X.

- 45 **8.** El procedimiento de la reivindicación 5, en el que:

la pluralidad de conjuntos de prioridad predefinidos incluye un primer conjunto de prioridad y un segundo conjunto de prioridad, y

50 el primer conjunto de prioridad y el segundo conjunto de prioridad se definen de modo que, para cada bloque vecino respectivo de la pluralidad de bloques vecinos:

el bloque vecino respectivo está en el primer conjunto de prioridad si el bloque vecino respectivo está codificado en un intermodo afín; y

55 el bloque vecino respectivo está en el segundo conjunto de prioridad si el bloque vecino respectivo está codificado en un modo de fusión afín.

- 60 **9.** El procedimiento de la reivindicación 1 o la reivindicación 2, en el que el bloque afín de origen es un primer bloque de origen afín, comprendiendo además el procedimiento:

seleccionar un segundo bloque afín de origen, siendo el segundo bloque afín de origen un bloque codificado por afinidad diferente que es vecino espacialmente al bloque actual;

65 extrapolar vectores de movimiento de los puntos de control del segundo bloque afín de origen para determinar segundos predictores de vectores de movimiento para los puntos de control del bloque actual; e

insertar un segundo conjunto de MVP afín en la lista de candidatos de conjunto de MVP afines, incluyendo el segundo conjunto de MVP afín los segundos predictores de vector de movimiento para los puntos de control del bloque actual.

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

una memoria configurada para almacenar los datos de vídeo; y

uno o más circuitos de procesamiento configurados para:

seleccionar un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es vecino espacialmente a un bloque actual;

extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual en base a los vectores de movimiento extrapolados;

insertar, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afín que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual;

determinar, en base a un índice señalizado en un flujo de bits, un conjunto de MVP afín seleccionado en la lista de candidatos de conjunto de MVP afines;

obtener, de un flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y los predictores de vectores de movimiento en el conjunto de MVP afín seleccionado;

determinar, en base a los predictores de vector de movimiento incluidos en el conjunto de MVP afín seleccionado y las MVD, los vectores de movimiento de los puntos de control del bloque actual;

generar, en base a los vectores de movimiento de los puntos de control del bloque actual, un bloque predictivo; y

reconstruir el bloque actual basado en datos residuales y el bloque predictivo.

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

una memoria configurada para almacenar los datos de vídeo; y

uno o más circuitos de procesamiento configurados para:

seleccionar un bloque afín de origen, siendo el bloque afín de origen un bloque codificado por afinidad que es vecino espacialmente a un bloque actual;

extrapolar vectores de movimiento de puntos de control del bloque afín de origen para determinar predictores de vector de movimiento para puntos de control del bloque actual en base a los vectores de movimiento extrapolados;

insertar, en una lista de candidatos de conjunto de predictores de vectores de movimiento afines (MVP), un conjunto de MVP afín que incluye los predictores de vectores de movimiento para los puntos de control del bloque actual;

seleccionar un conjunto de MVP afín en la lista de candidatos de conjunto de MVP afines;

señalizar, en un flujo de bits, diferencias de vector de movimiento (MVD) que indican diferencias entre vectores de movimiento de los puntos de control del bloque actual y los predictores de vectores de movimiento en el conjunto de MVP afín seleccionado; y

señalizar, en el flujo de bits, un índice que indica una posición en la lista de candidatos de conjunto de MVP afines del conjunto de MVP afín seleccionado.

12. El dispositivo de la reivindicación 10 o la reivindicación 11, en el que los puntos de control del bloque actual incluyen un primer punto de control y un segundo punto de control, los uno o más circuitos de procesamiento configurados además para:

- determinar un primer predictor de vector de movimiento como un vector de movimiento de un bloque adyacente al primer punto de control del bloque actual;
- 5 determinar un segundo predictor de vector de movimiento como un vector de movimiento de un bloque adyacente al segundo punto de control del bloque actual; e
- insertar, en la lista de candidatos de conjunto de MVP afines, un conjunto de MVP afín que incluye el primer predictor de vector de movimiento y el segundo predictor de vector de movimiento.
- 10 **13.** El dispositivo de la reivindicación 10 o la reivindicación 11, en el que el bloque actual es adyacente a una pluralidad de bloques vecinos, y los uno o más circuitos de procesamiento están configurados de modo que, como parte de la selección del bloque afín de origen, los uno o más circuitos de procesamiento:
- 15 determinan que el bloque afín de origen es un bloque codificado por afinidad que aparece primero de la pluralidad de bloques vecinos visitados en un orden de visita predefinido.
- 14.** El dispositivo de la reivindicación 10 o la reivindicación 11, en el que el bloque actual es adyacente a una pluralidad de bloques vecinos, y los uno o más circuitos de procesamiento están configurados de modo que, como parte de la selección del bloque afín de origen, los uno o más circuitos de procesamiento:
- 20 determinan que el bloque afín de origen es un bloque codificado por afinidad disponible que aparece primero de la pluralidad de bloques vecinos de acuerdo con una pluralidad de conjuntos de prioridad predefinidos en base a un orden de visita predefinido, en el que el bloque afín de origen no se considera disponible si el bloque afín de origen no está en uno de los conjuntos de prioridad predefinidos.
- 25 **15.** Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, provocan que uno o más circuitos de procesamiento de un dispositivo realicen el procedimiento de acuerdo con una cualquiera de las reivindicaciones 1 a 9.

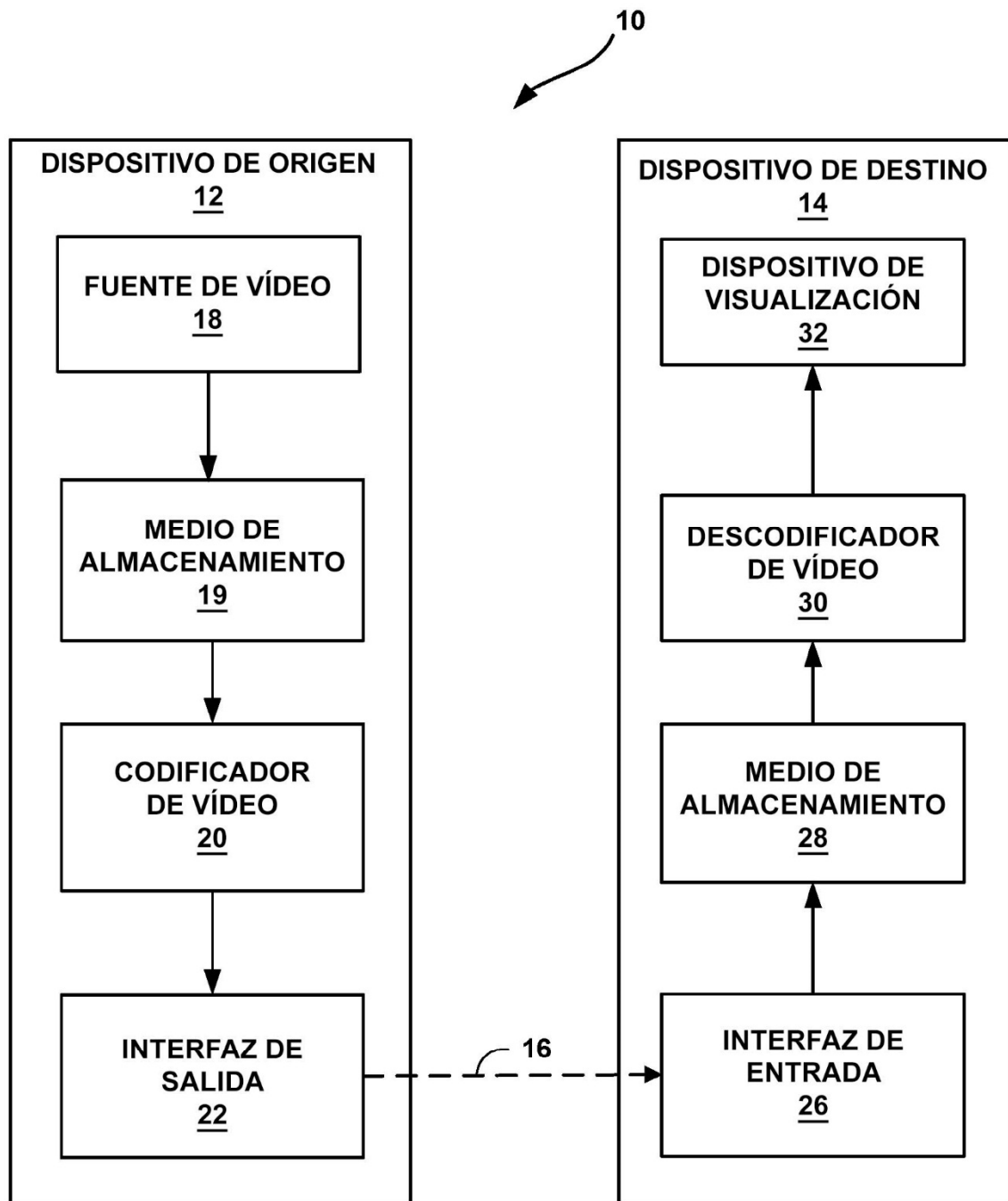
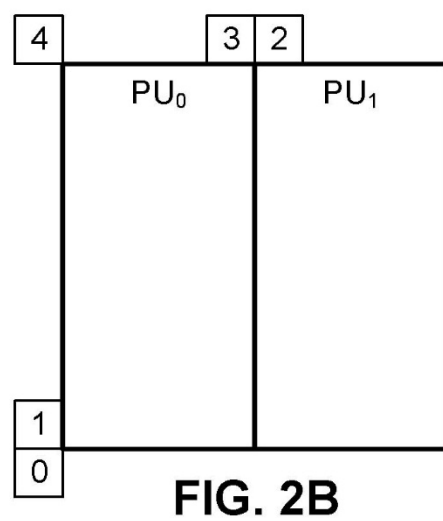
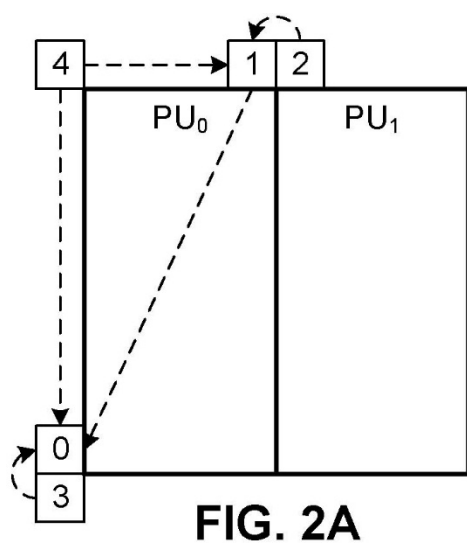


FIG. 1



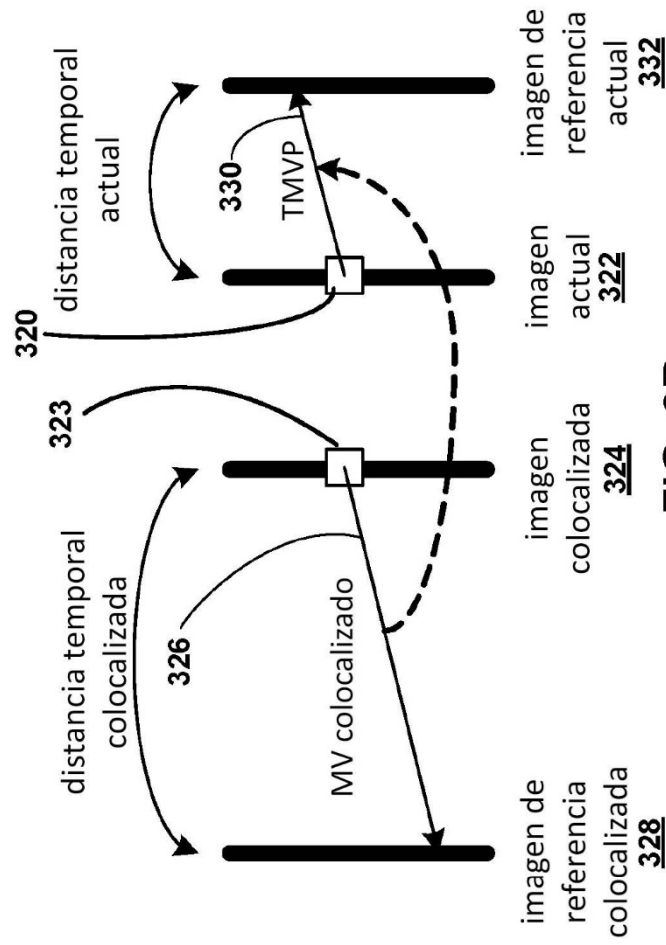


FIG. 3B

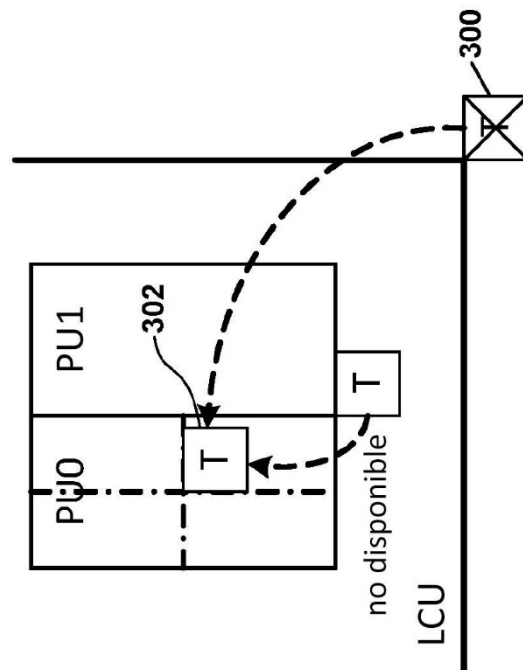


FIG. 3A

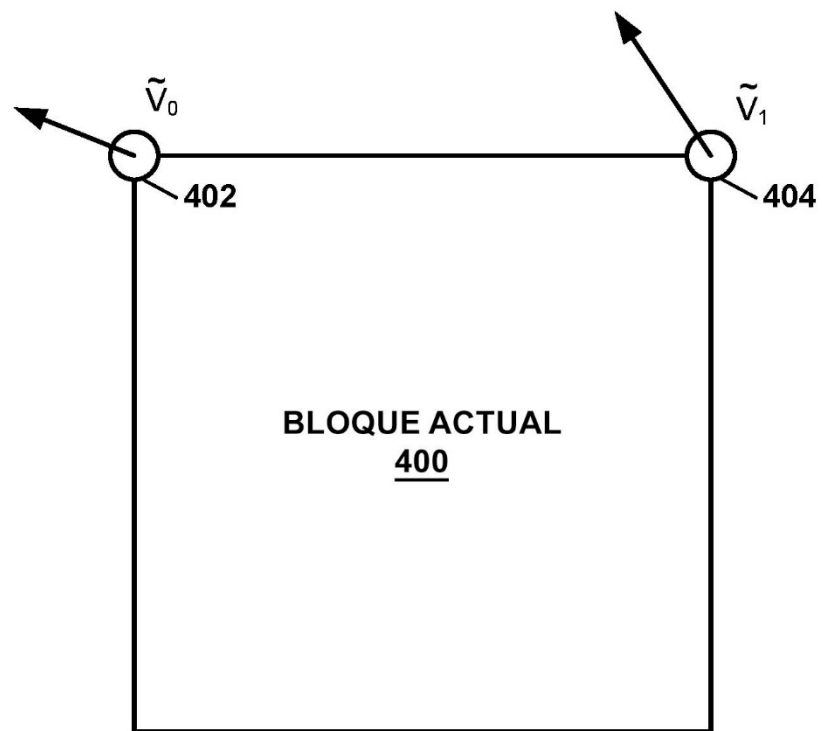


FIG. 4

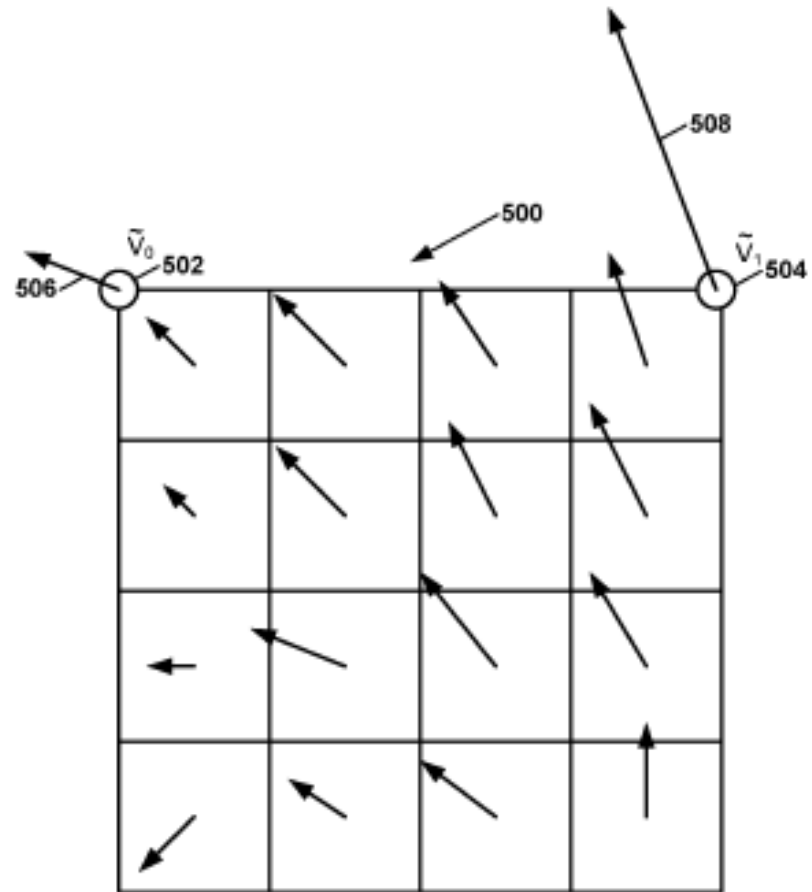


FIG. 5

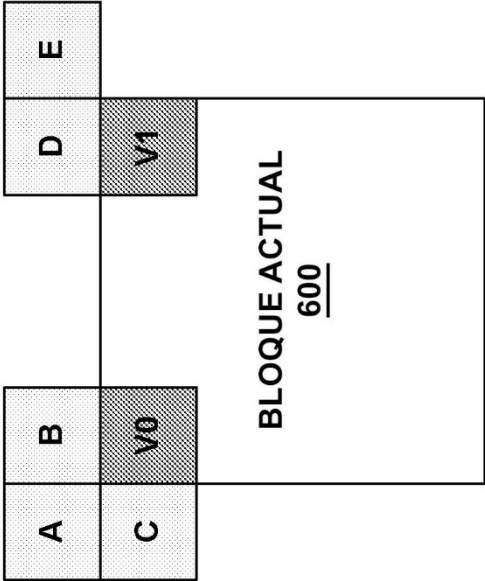


FIG. 6A

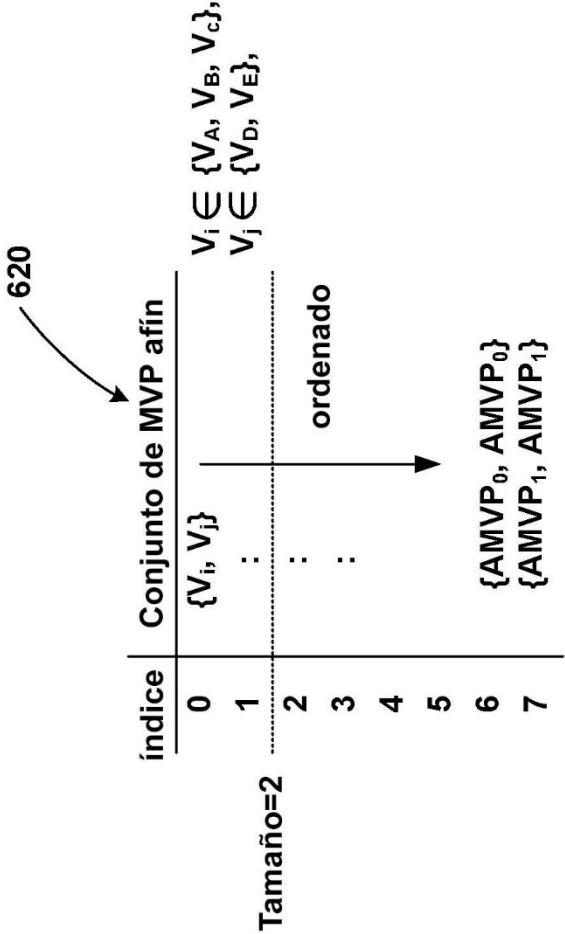


FIG. 6B

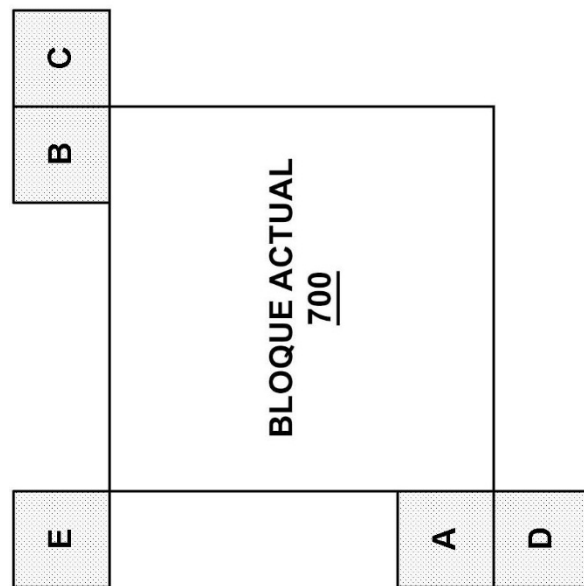


FIG. 7A

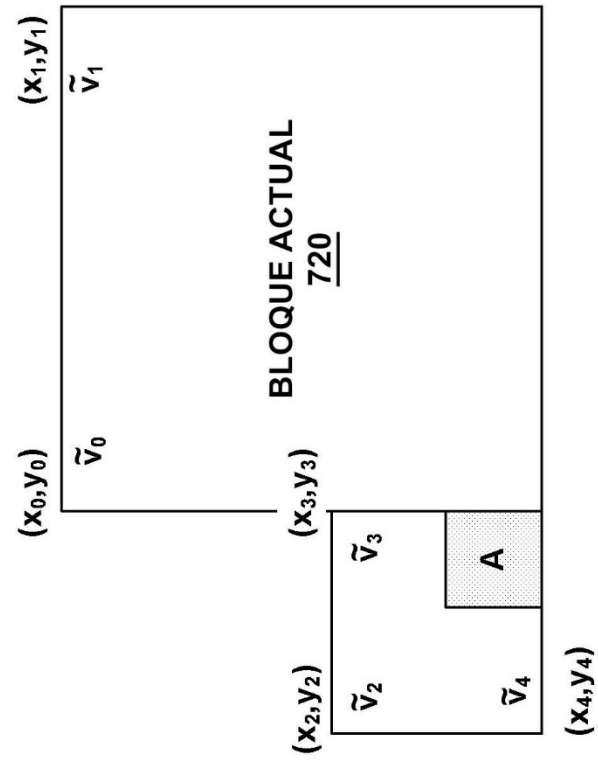


FIG. 7B

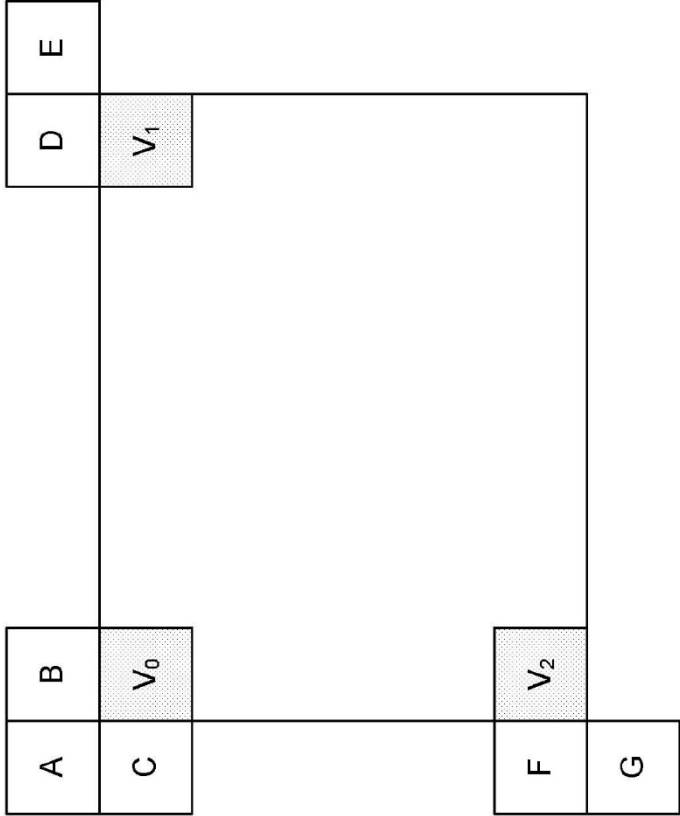


FIG. 8A

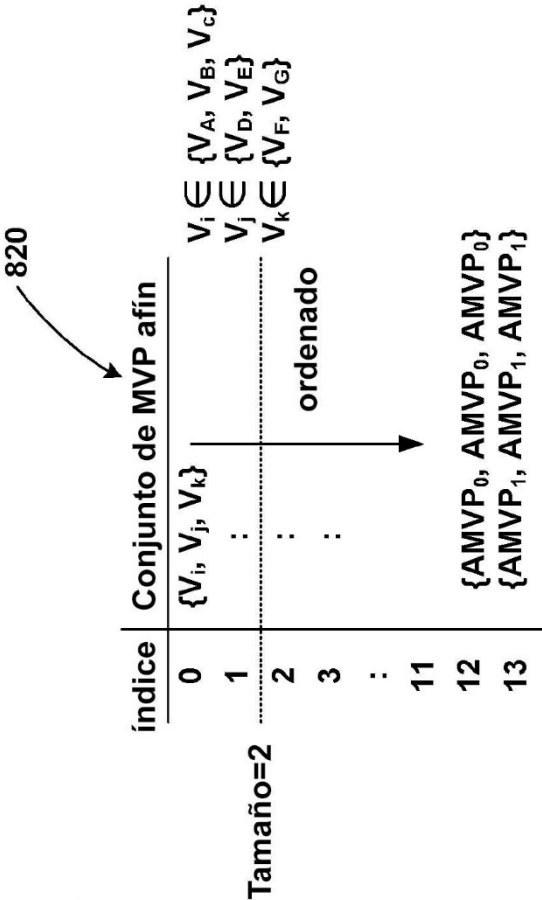


FIG. 8B

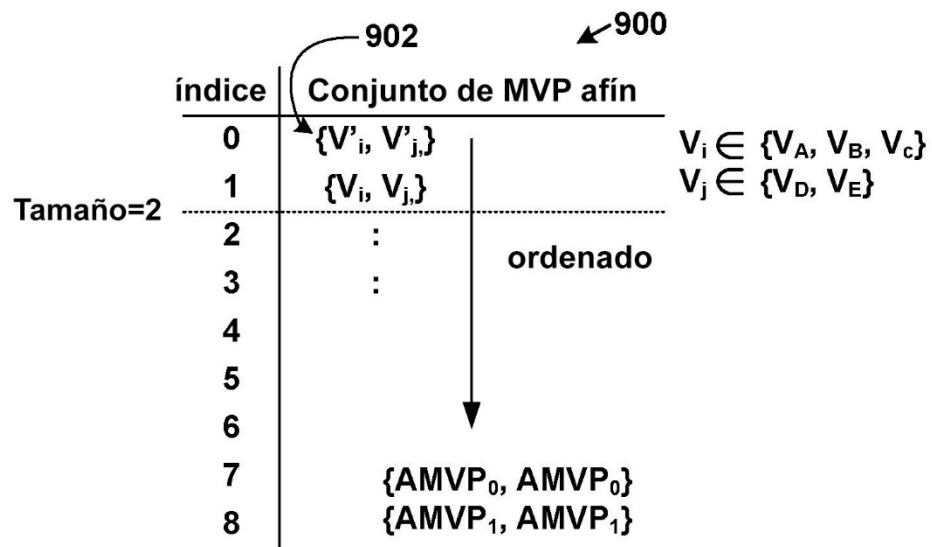


FIG. 9

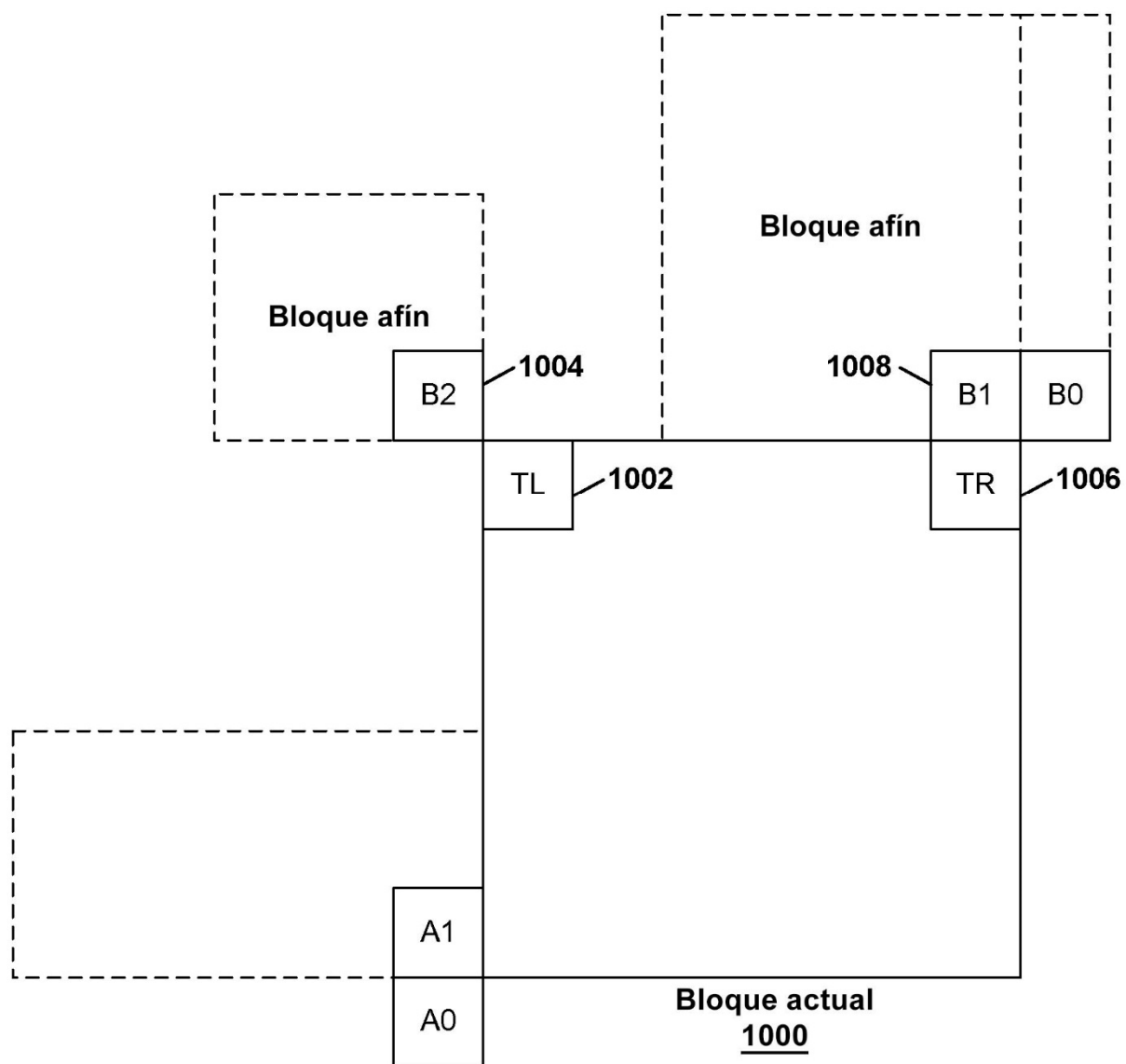


FIG. 10

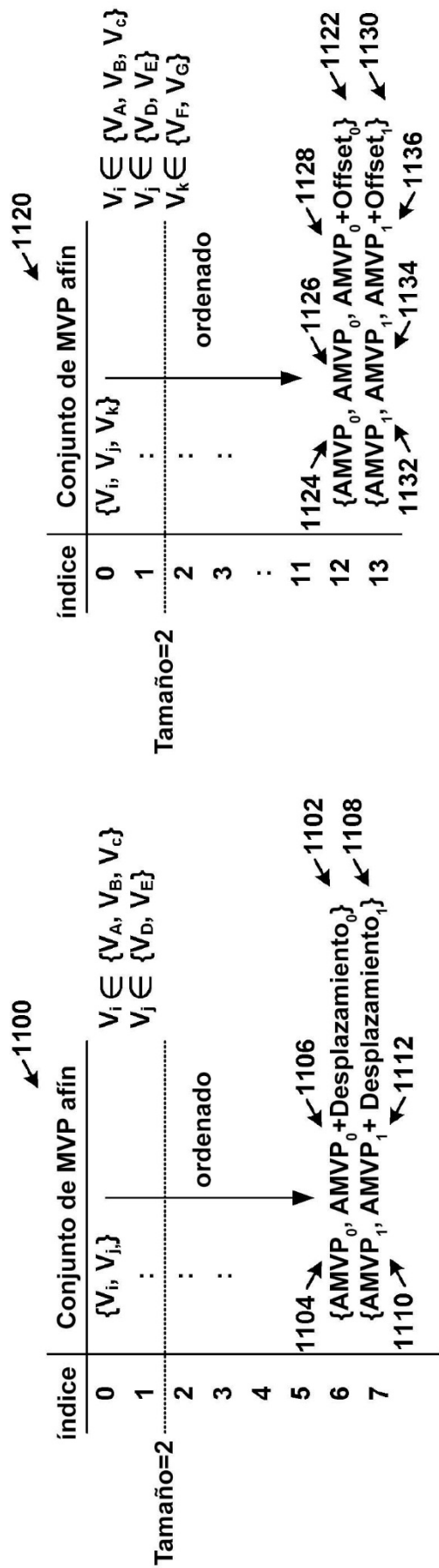
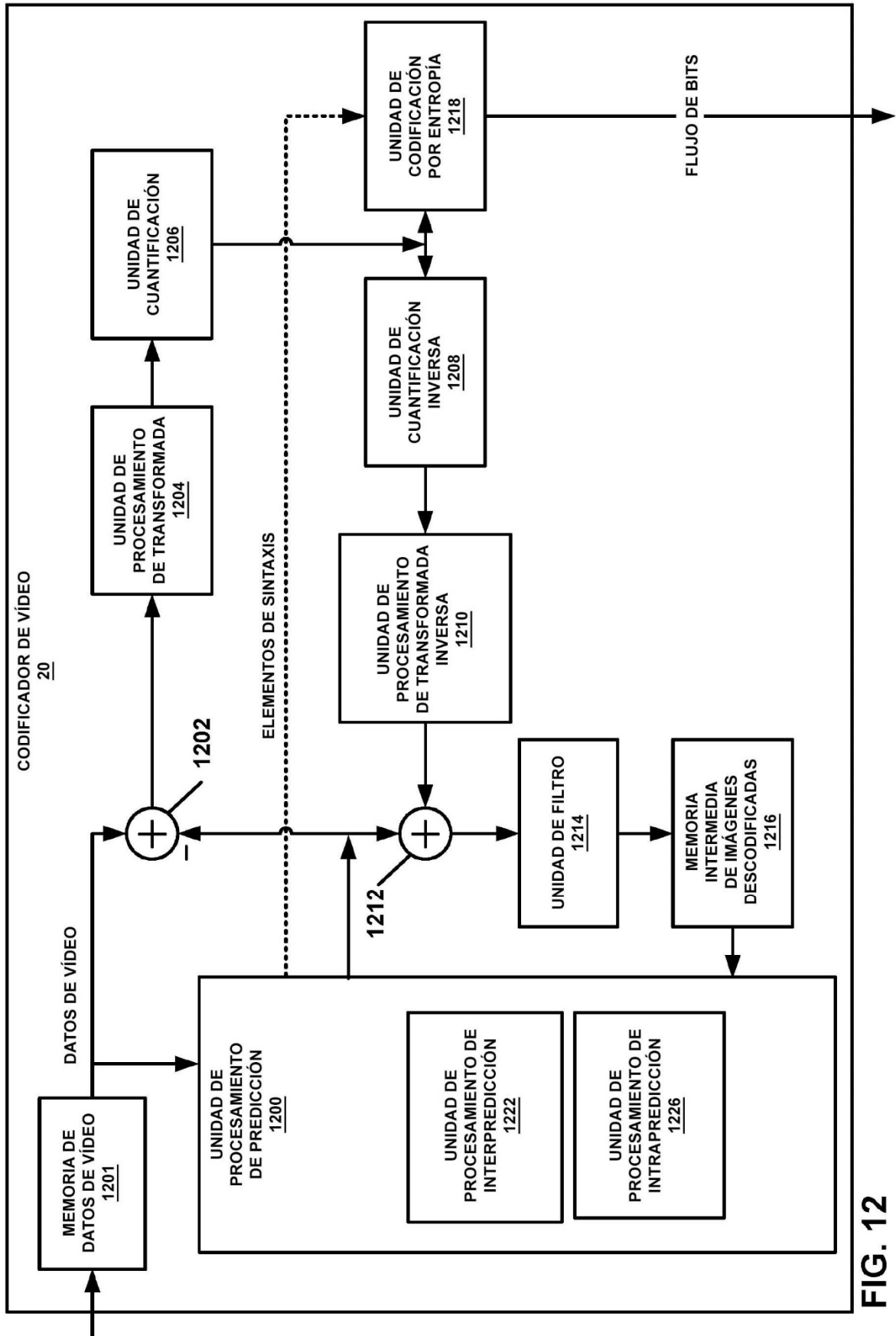


FIG. 11A

FIG. 11B



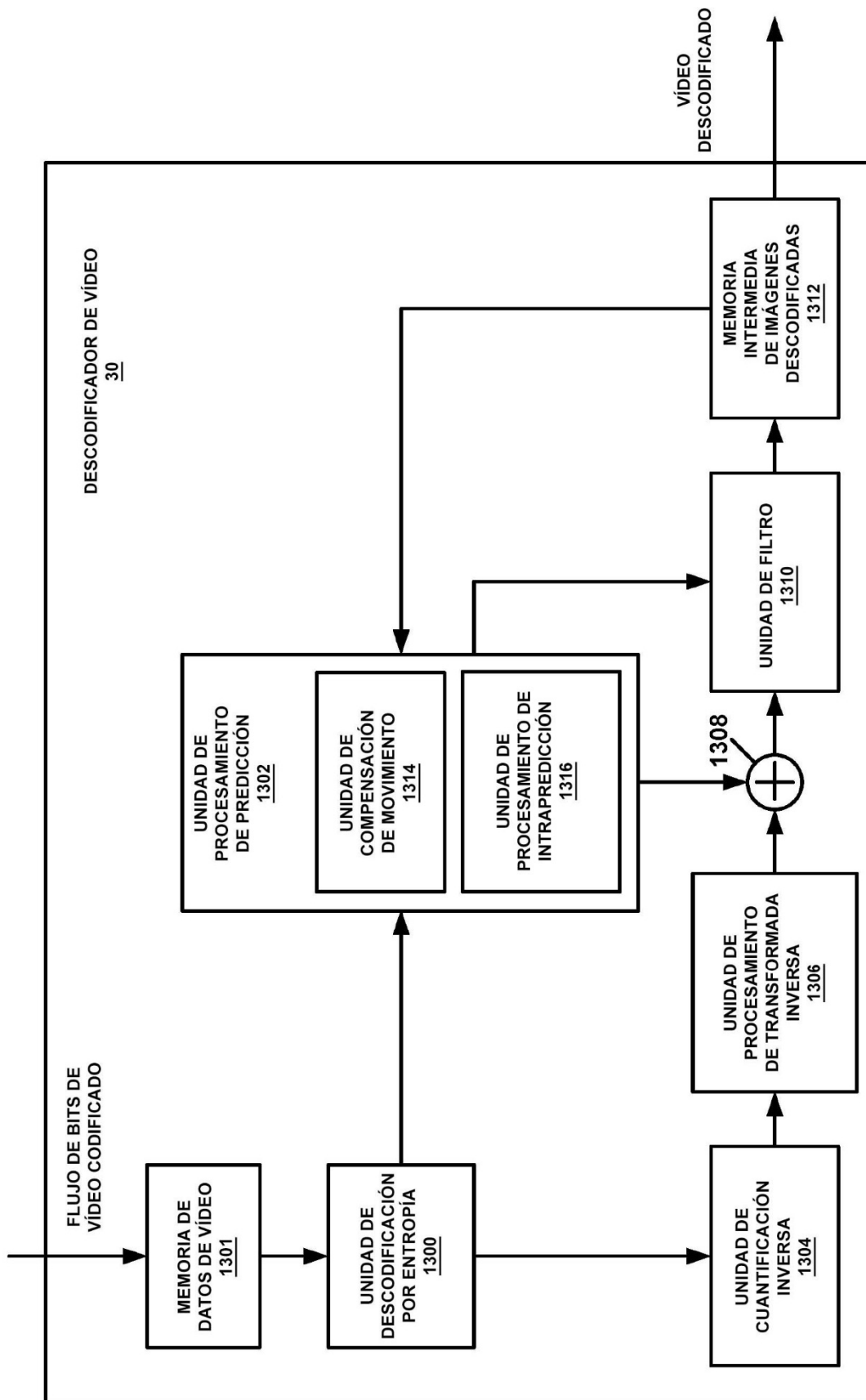


FIG. 13

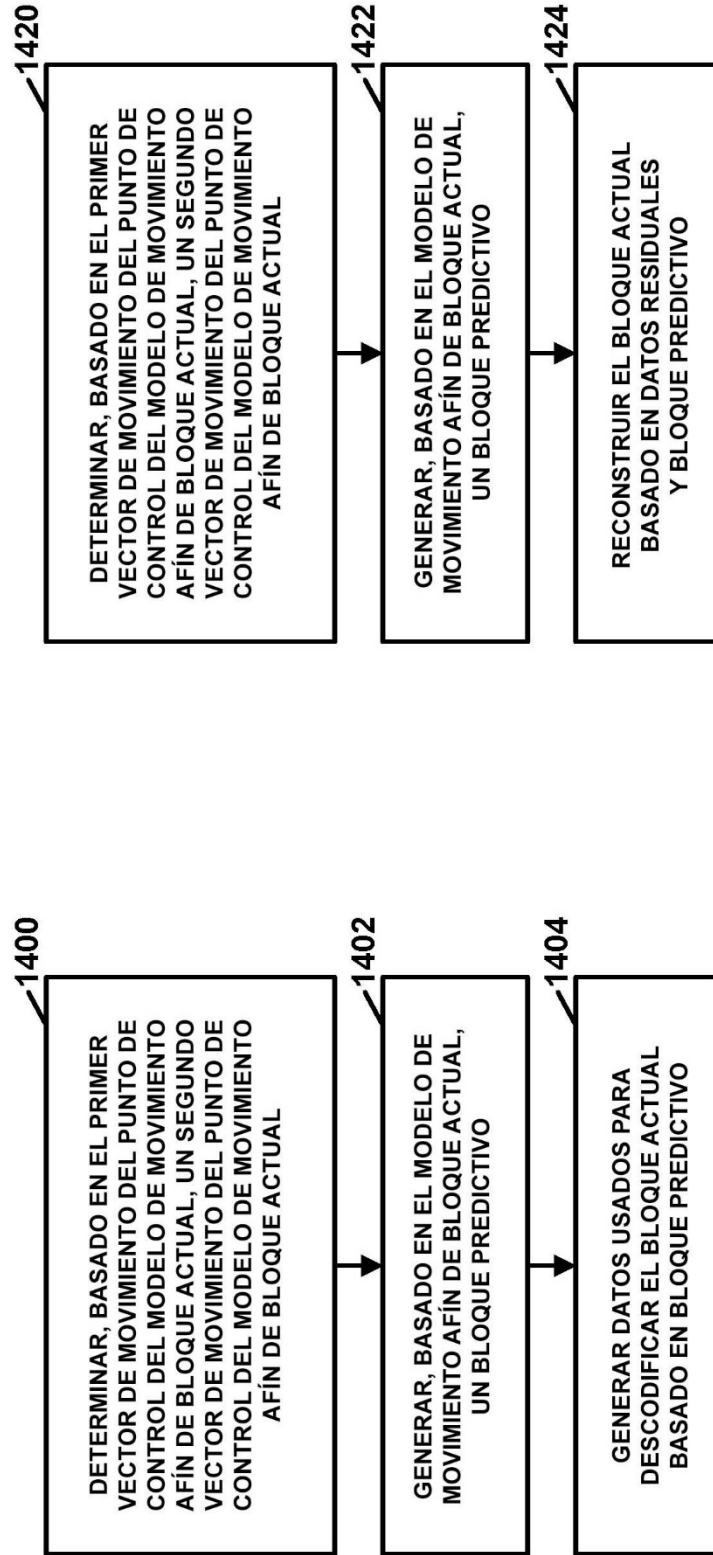
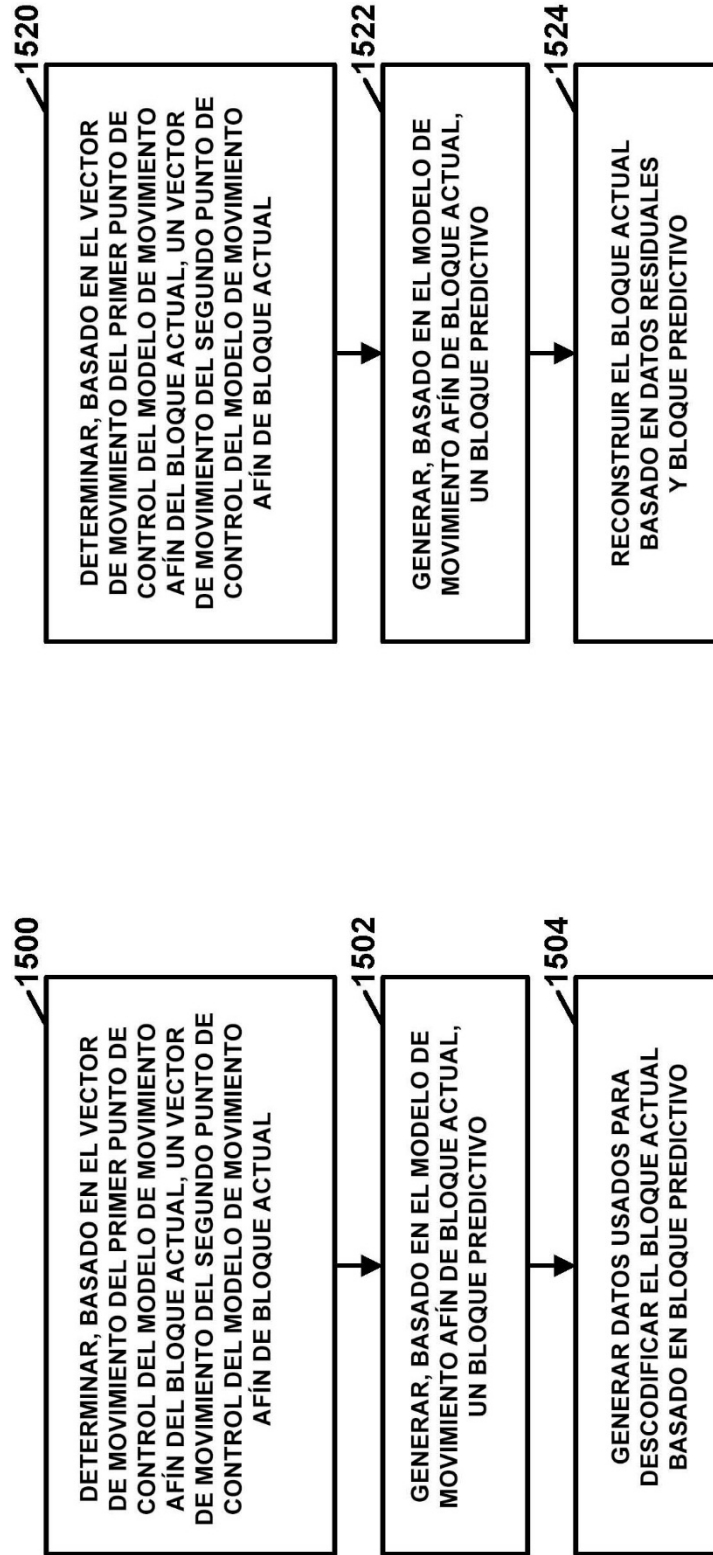
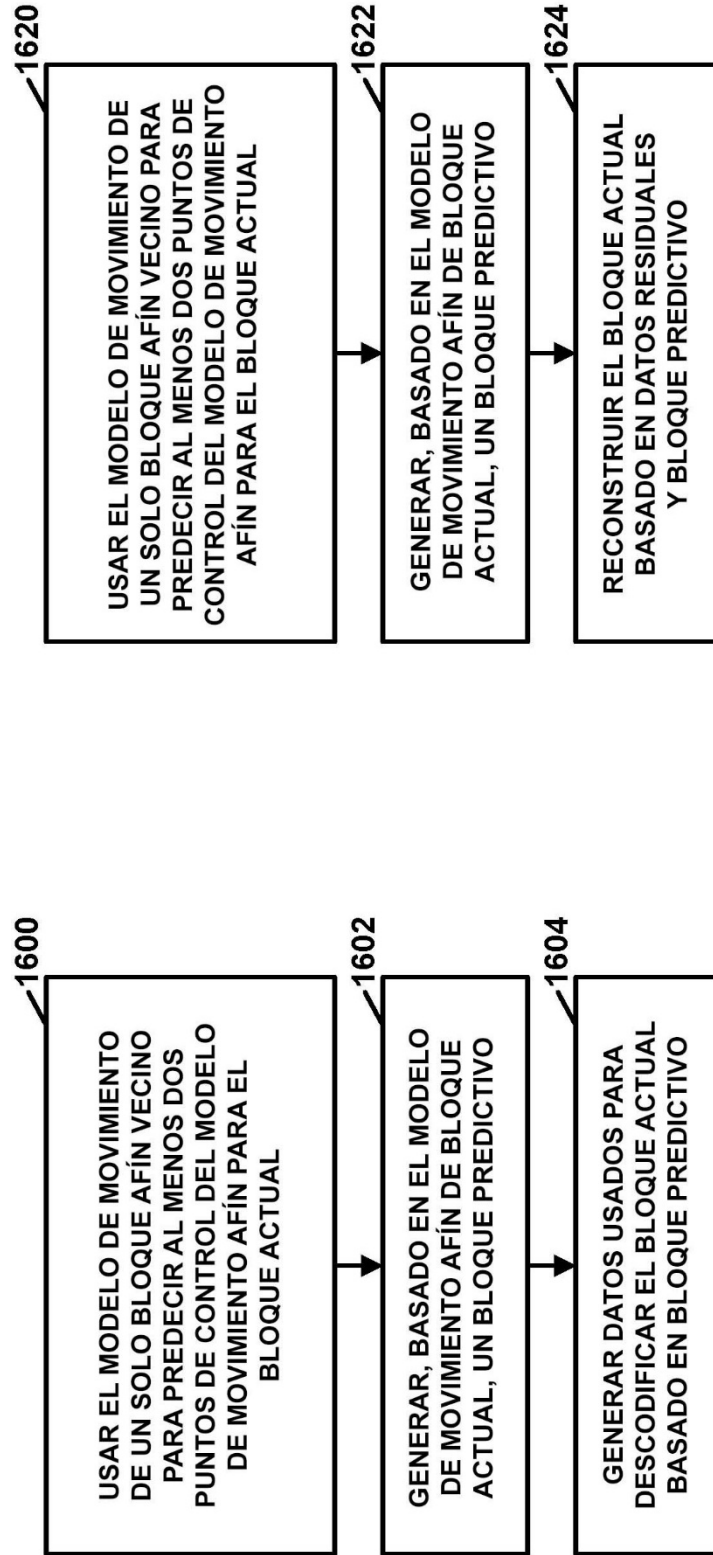


FIG. 14A

FIG. 14B





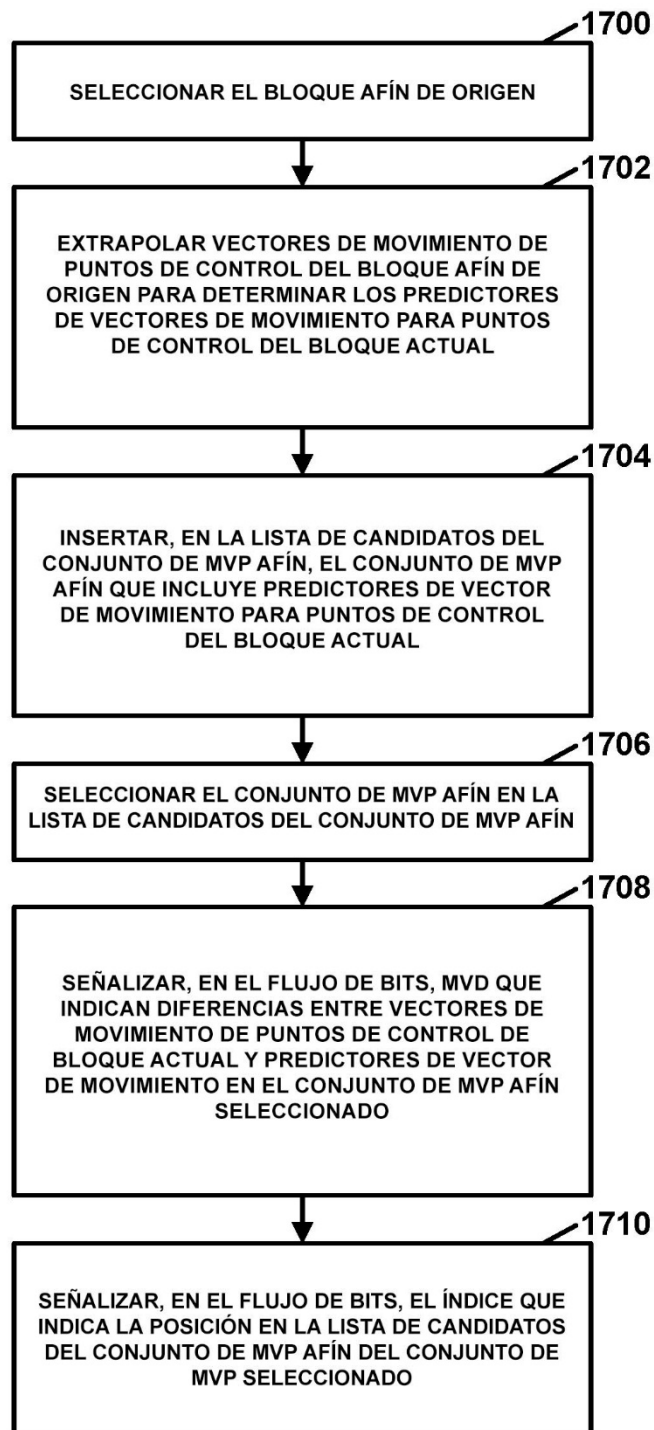


FIG. 17

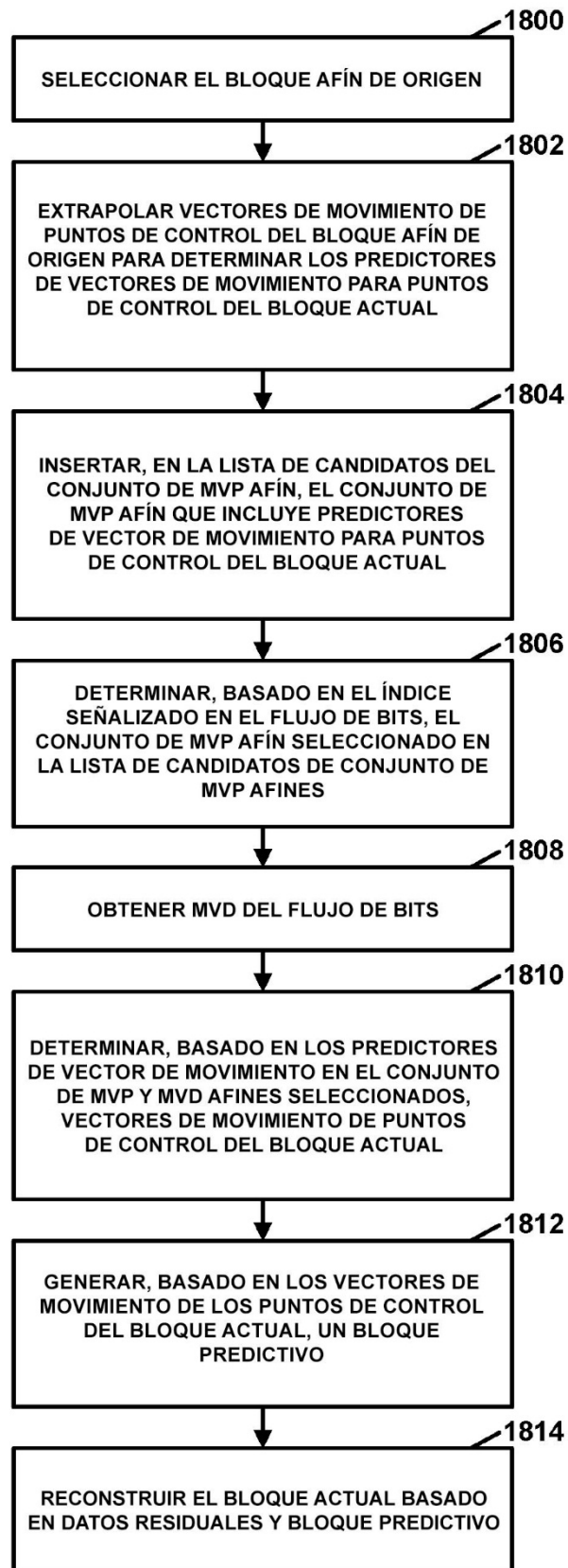


FIG. 18

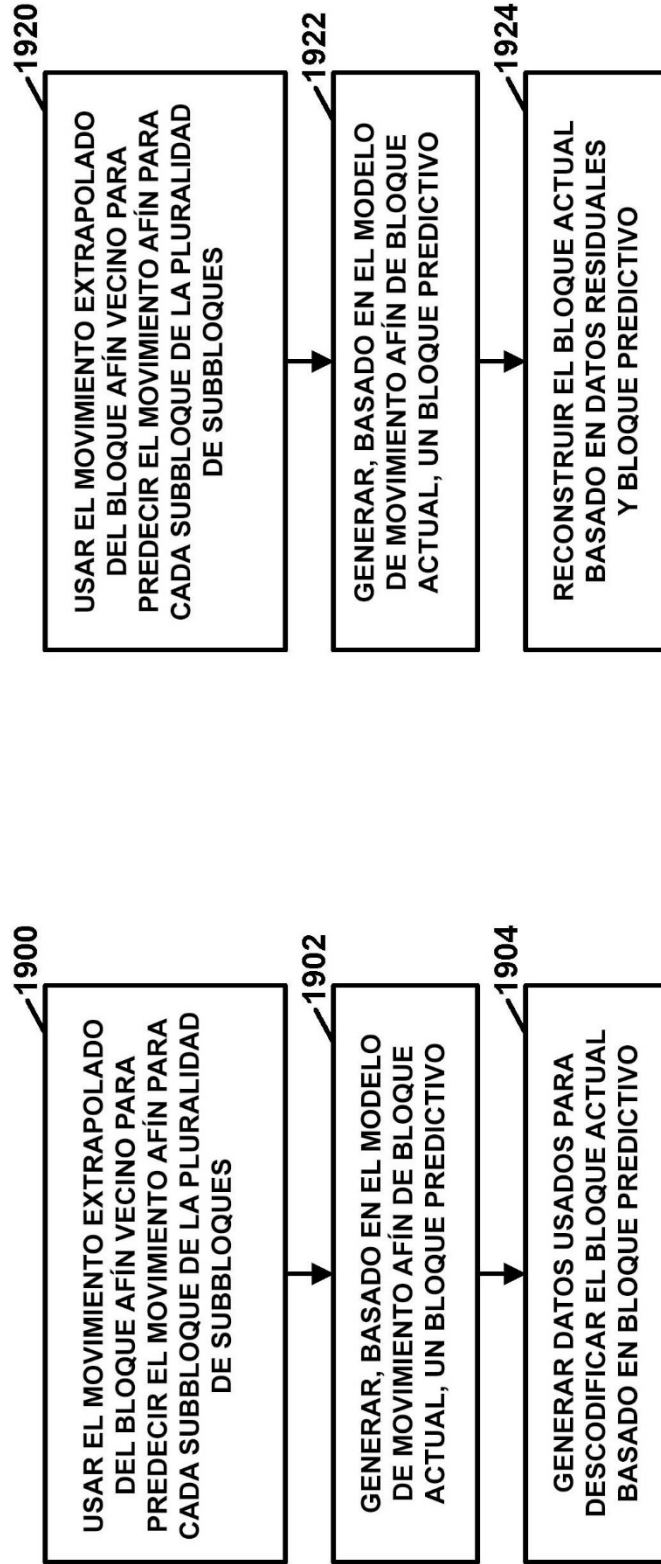


FIG. 19A

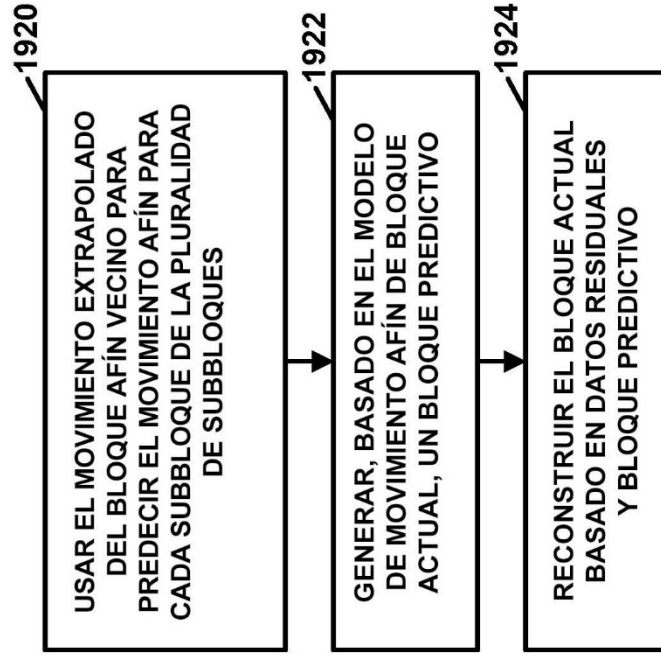


FIG. 19B

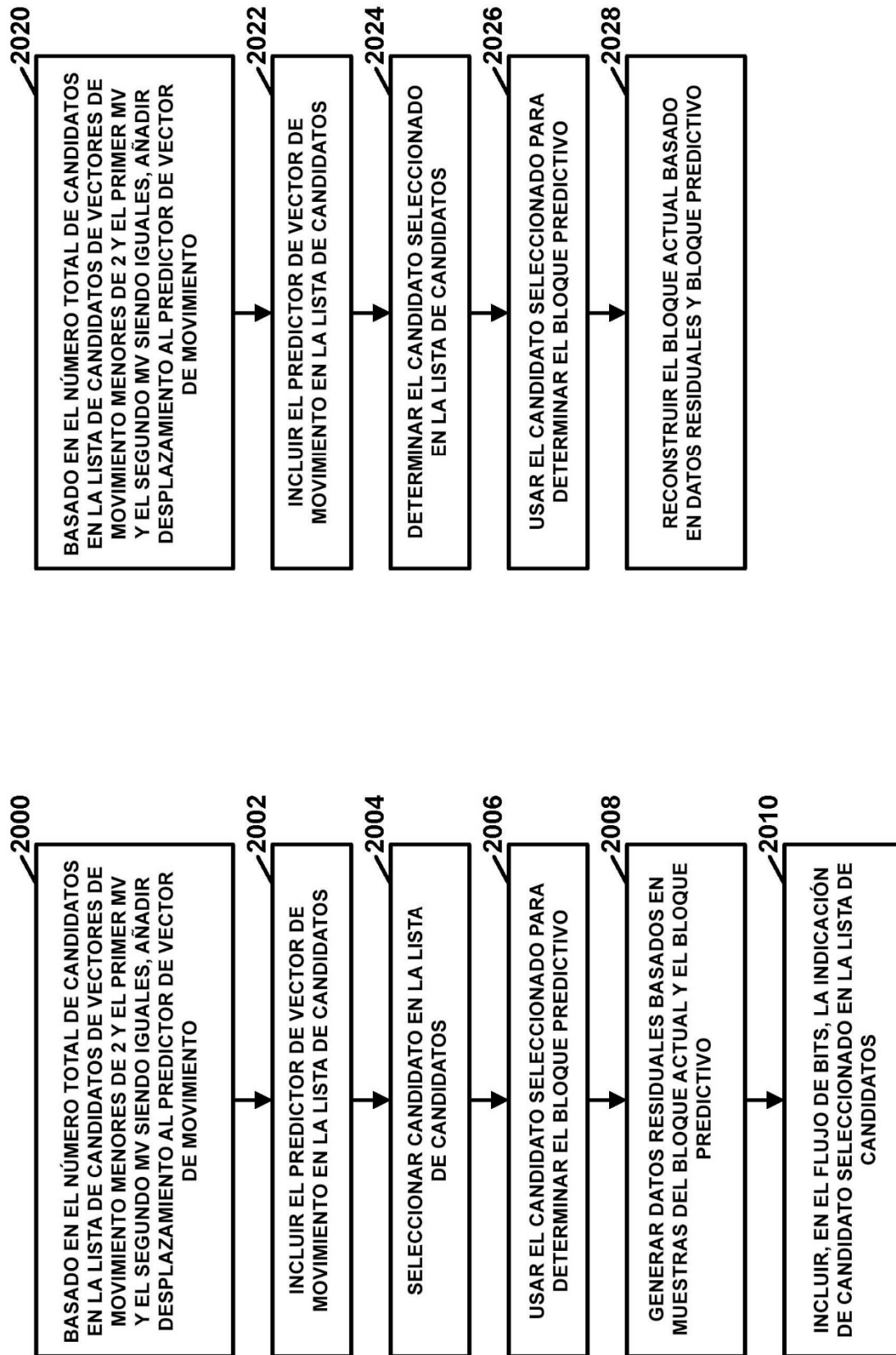


FIG. 20B

FIG. 20A

