

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 633 150**

51 Int. Cl.:

H04N 19/597 (2014.01)
H04N 19/105 (2014.01)
H04N 19/52 (2014.01)
H04N 19/176 (2014.01)
H04N 19/147 (2014.01)
H04N 19/46 (2014.01)
H04N 19/513 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **20.04.2012 PCT/US2012/034516**
- 87 Fecha y número de publicación internacional: **26.10.2012 WO12145670**
- 96 Fecha de presentación y número de la solicitud europea: **20.04.2012 E 12718013 (1)**
- 97 Fecha y número de publicación de la concesión europea: **12.04.2017 EP 2700228**

54 Título: **Predicción de vectores de movimiento en codificación de vídeo**

30 Prioridad:

20.04.2011 US 201161477561 P
28.07.2011 US 201161512765 P
19.04.2012 US 201213451204

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
19.09.2017

73 Titular/es:

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

72 Inventor/es:

CHEN, YING;
CHEN, PEISONG y
KARCZEWICZ, MARTA

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 633 150 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 en codificación de vídeo

5 Descripción

Esta divulgación reivindica el beneficio de prioridad de la solicitud provisional de Estados Unidos n.º 61/477.561, presentada el 20 de abril de 2011, y de la solicitud de Estados Unidos n.º 61/512.765, presentada el 28 de julio de 2011.

10 CAMPO TÉCNICO

Esta divulgación se refiere a la codificación de vídeo.

15 ANTECEDENTES

Las capacidades de vídeo digital pueden incorporarse en una amplia gama de dispositivos, que incluye televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo en continuo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación avanzada de vídeo (AVC), la norma de Codificación de vídeo de alta eficacia (HEVC) actualmente en elaboración y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de compresión de vídeo.

Las técnicas de compresión de vídeo llevan a cabo la predicción espacial (intraimagen) y/o la predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una imagen o una parte de una imagen) puede dividirse en bloques de vídeo, que también pueden denominarse bloques en árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un segmento sometido a codificación intra (I) de una imagen se codifican mediante predicción espacial, con respecto a unas muestras de referencia de unos bloques contiguos de la misma imagen. Los bloques de vídeo de un segmento sometido a codificación inter (P o B) de una imagen pueden usar la predicción espacial con respecto a unas muestras de referencia de bloques contiguos de la misma imagen, o la predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia.

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque por codificar. Los datos residuales representan diferencias de píxeles entre el bloque original por codificar y el bloque predictivo. Un bloque sometido a codificación inter 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 sometido a codificación intra se codifica de acuerdo con una modalidad de codificación intra y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse del dominio del píxel a un dominio de transformada, dando como resultado unos coeficientes de transformada residuales, los cuales pueden cuantificarse posteriormente. Los coeficientes de transformada cuantificados, inicialmente dispuestos en una matriz bidimensional, pueden explorarse con el fin de producir un vector unidimensional de coeficientes de transformada, y puede aplicarse codificación de entropía para lograr aún más compresión.

El documento WO 2008/007913 A1 divulga un procedimiento para predecir un vector de movimiento de dirección temporal para una unidad actual que usa un vector de movimiento de dirección de vista asociado con la unidad actual para identificar una unidad de referencia en una vista diferente.

El documento EP 1 377 067 A1 divulga un procedimiento en el que los vectores de movimiento se obtienen como vectores predictivos escalando los vectores de referencia de acuerdo con las distancias intratrama.

SUMARIO

En general, esta divulgación describe técnicas para codificación de datos de vídeo. Esta divulgación describe técnicas para realizar la predicción de vectores de movimiento, la estimación de movimiento y la compensación de movimiento cuando se realiza la codificación intermodalidad (es decir, se codifica un bloque actual en relación con bloques de otras imágenes) en la codificación de vídeo multivista (MVC). En general, la MVC es una norma de codificación de vídeo para encapsular múltiples vistas de datos de vídeo. Cada vista puede corresponder a una perspectiva, o ángulo, diferente, donde se captaron los datos de vídeo correspondientes de una escena común. Las técnicas de esta divulgación incluyen en general la predicción de datos de predicción de movimiento en el contexto

de la codificación de vídeo multivista. Es decir, por ejemplo, de acuerdo con las técnicas de esta divulgación, un vector de movimiento de disparidad, de un bloque de una vista igual o diferente a la de un bloque que se está codificando actualmente, puede usarse para predecir el vector de movimiento del bloque actual. En otro ejemplo, de acuerdo con las técnicas de esta divulgación, un vector de movimiento temporal de un bloque de una vista igual o diferente a la de un bloque codificado actualmente puede usarse para predecir el vector de movimiento del bloque actual.

En un ejemplo, unos aspectos de esta divulgación se refieren a un procedimiento de codificación de datos de vídeo de acuerdo con la reivindicación independiente 1.

En otro ejemplo, unos aspectos de esta divulgación se refieren a un aparato para codificar datos de vídeo de acuerdo con la reivindicación independiente 7.

En otro ejemplo, unos aspectos de esta divulgación se refieren a un medio de almacenamiento legible por ordenador de acuerdo con la reivindicación independiente 9.

Los detalles de uno o más aspectos de la divulgación se exponen en los dibujos adjuntos y la descripción siguiente. Otras características, objetos y ventajas de las técnicas descritas en esta divulgación serán evidentes a partir de la descripción y los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

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

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

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

La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de patrón de predicción de codificación de vídeo multivista (MVC).

La FIG. 5 es un diagrama de bloques que ilustra ejemplos de localizaciones para predictores de vector de movimiento candidatos.

La FIG. 6 es un diagrama conceptual que ilustra la generación y el escalado de un predictor de vector de movimiento, de acuerdo con unos aspectos de esta divulgación.

La FIG. 7 es otro diagrama conceptual que ilustra la generación y el escalado de un predictor de vector de movimiento, de acuerdo con unos aspectos de esta divulgación.

La FIG. 8 es otro diagrama conceptual que ilustra la generación y el escalado de un predictor de vector de movimiento, de acuerdo con unos aspectos de esta divulgación.

La FIG. 9 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de información de predicción para un bloque de datos de vídeo.

La FIG. 10 es un diagrama conceptual que ilustra la generación de un predictor de vector de movimiento a partir de un bloque en una vista diferente a la de un bloque actual.

La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de procedimiento de generación de un predictor de vector de movimiento a partir de un bloque en una vista diferente a la de un bloque actual.

DESCRIPCIÓN DETALLADA

De acuerdo con ciertos sistemas de codificación de vídeo, la estimación de movimiento y la compensación de movimiento se pueden usar para reducir la redundancia temporal en una secuencia de vídeo, con el fin de lograr la compresión de datos. En este caso, puede generarse un vector de movimiento que identifica un bloque predictivo de datos de vídeo, por ejemplo, un bloque de otra imagen o segmento de vídeo, que se puede usar para predecir los valores del bloque de vídeo actual que se está codificando. Los valores del bloque de vídeo predictivo se restan de los valores del bloque de vídeo actual para producir un bloque de datos residuales. La información de movimiento (por ejemplo, un vector de movimiento, índices de vectores de movimiento, direcciones de predicción u otra información) se comunica desde un codificador de vídeo a un decodificador de vídeo, junto con los datos residuales. El decodificador puede localizar el mismo bloque predictivo (basándose en el vector de movimiento) y reconstruir el

bloque de vídeo codificado combinando los datos residuales con los datos del bloque predictivo.

En algunos casos, la codificación predictiva de vectores de movimiento también se aplica para reducir aún más la cantidad de datos necesarios para comunicar el vector de movimiento. El establecimiento de un vector de movimiento se realiza desde una imagen objetivo hasta una imagen de referencia. Un vector de movimiento puede predecirse espacialmente o temporalmente. Un vector de movimiento predicho espacialmente está asociado con bloques espaciales disponibles (un bloque del mismo instante de tiempo). Un vector de movimiento predicho temporalmente está asociado con bloques temporales disponibles (un bloque de un instante de tiempo diferente). En el caso de la predicción del vector de movimiento, en lugar de codificar y comunicar el propio vector de movimiento, el codificador codifica y comunica una diferencia de vector de movimiento (MVD) con respecto a un vector de movimiento conocido (o conocible). En la norma H.264/AVC, el vector de movimiento conocido, que puede usarse con la MVD para definir el vector de movimiento actual, puede definirse mediante uno de los denominados predictores de vector de movimiento (MVP). Para ser un MVP válido, el vector de movimiento debe apuntar a la misma imagen que el vector de movimiento que se está codificando actualmente mediante el MVP y la MVD.

Un codificador de vídeo puede construir una lista de posibles predictores de vector de movimiento que incluye varios bloques vecinos en direcciones espaciales y temporales como MVP candidatos. En este caso, un codificador de vídeo puede seleccionar el predictor más preciso del conjunto candidato basándose en el análisis de la velocidad de codificación y la distorsión (por ejemplo, mediante un análisis velocidad-coste de distorsión u otro tipo de análisis de eficiencia de codificación). Se puede transmitir un índice de predictor de vector de movimiento (`mvp_idx`) a un decodificador de vídeo para informar al decodificador dónde localizar el MVP. El MVD también se transmite. El decodificador puede combinar la MVD con el MVP (definido por el índice del predictor del vector de movimiento) para reconstruir el vector de movimiento.

También puede estar disponible una denominada "modalidad de fusión", en la que se hereda información de movimiento (tal como vectores de movimiento, índices de imágenes de referencia, direcciones de predicción u otra información) de un bloque de vídeo vecino para un bloque de vídeo actual que se está codificando. Se puede usar un valor de índice para identificar el vecino del cual el bloque de vídeo actual hereda su información de movimiento.

La codificación de vídeo multivista (MVC) es una norma de codificación de vídeo para encapsular múltiples vistas de datos de vídeo. En general, cada vista se corresponde con una perspectiva, o ángulo, diferente, donde se captaron (o generaron) los datos de vídeo correspondientes de una escena común. La MVC proporciona un conjunto de metadatos, es decir, datos descriptivos para las vistas, de forma conjunta o individual.

Las vistas codificadas se pueden usar para la visualización tridimensional (3D) de datos de vídeo. Por ejemplo, dos vistas (por ejemplo, las vistas de los ojos izquierdo y derecho de un observador humano) pueden visualizarse simultáneamente, o casi simultáneamente, mediante diferentes polarizaciones de luz, y un espectador puede usar gafas polarizadas pasivas de tal manera que cada uno de los ojos del espectador reciba una respectiva de las vistas. De forma alternativa, el espectador puede llevar gafas activas que obturan cada ojo de forma independiente, y una pantalla puede alternar rápidamente entre las imágenes de cada ojo de forma sincronizada con las gafas.

En la MVC, una imagen particular de una vista particular se conoce como "componente de vista". Es decir, un componente de vista de una vista corresponde a un instante de tiempo particular de la vista. Típicamente, los objetos iguales o correspondientes de dos vistas no están coubicados. El término "vector de disparidad" puede usarse para referirse a un vector que indica el desplazamiento de un objeto en una imagen de una vista relativa al objeto correspondiente de una vista diferente. Dicho vector también se puede denominar "vector de desplazamiento". Un vector de disparidad también puede ser aplicable a un píxel o un bloque de datos de vídeo de una imagen. Por ejemplo, un píxel de una imagen de una primera vista puede estar desplazado con respecto a un píxel correspondiente de una imagen de una segunda vista por una disparidad particular relacionada con diferentes posiciones de la cámara desde las cuales se captan la primera vista y la segunda vista. En algunos ejemplos, la disparidad puede usarse para predecir un vector de movimiento de una vista a otra.

En el contexto de la MVC, las imágenes de una vista se pueden predecir a partir de las imágenes de otra vista. Por ejemplo, un bloque de datos de vídeo puede predecirse con relación a un bloque de datos de vídeo de una imagen de referencia del mismo instante de tiempo, pero de una vista diferente. En un ejemplo, un bloque que se está codificando actualmente puede denominarse "bloque actual". Un vector de movimiento que predice el bloque actual a partir de un bloque de una vista diferente pero del mismo instante de tiempo se denomina "vector de movimiento de disparidad". Un vector de movimiento de disparidad es típicamente aplicable en el contexto de la codificación de vídeo multivista, donde pueden estar disponibles más de una vista. De acuerdo con esta divulgación, una "distancia de vista" para un vector de movimiento de disparidad puede referirse a una diferencia de traslación entre la vista de la imagen de referencia y la vista de la imagen objetivo. Es decir, una distancia de vista puede representarse como una diferencia de identificador de vista entre un identificador de vista de la imagen de referencia y un identificador de vista de la imagen objetivo.

Otro tipo de vector de movimiento es un "vector de movimiento temporal". En el contexto de la codificación de vídeo multivista, un vector de movimiento temporal se refiere a un vector de movimiento que predice un bloque actual a

partir de un bloque de un instante de tiempo diferente, pero dentro de la misma vista. De acuerdo con esta divulgación, una "distancia temporal" de un vector de movimiento temporal puede referirse a una distancia de recuento de orden de imágenes (POC) desde la imagen de referencia a la imagen objetivo.

5 Ciertas técnicas de esta divulgación están dirigidas al uso de información de movimiento (por ejemplo, un vector de movimiento, unos índices de movimiento vectorial, unas direcciones de predicción u otra información) asociada con un bloque de datos de vídeo en una configuración multivista para predecir información de movimiento de un bloque que se está codificando actualmente. Por ejemplo, de acuerdo con unos aspectos de esta divulgación, se puede añadir un vector de movimiento predicho a partir de una vista diferente, como candidato para una o más listas de
10 vectores de movimiento usadas para la predicción del vector de movimiento del bloque actual. En algunos ejemplos, un codificador de vídeo puede usar un vector de movimiento de disparidad asociado con un bloque de una vista diferente a la de un bloque que se codifica actualmente para predecir un vector de movimiento para el bloque actual y puede añadir el vector de movimiento de disparidad predicho a una lista de vectores de movimiento candidatos. En otros ejemplos, un codificador de vídeo puede usar un vector de movimiento temporal asociado con un bloque de
15 una vista diferente a la de un bloque que se codifica actualmente para predecir un vector de movimiento para el bloque actual, y puede añadir el vector de movimiento temporal predicho a una lista de vectores de movimiento candidatos.

20 De acuerdo con unos aspectos de esta divulgación, un vector de movimiento de disparidad puede escalarse antes de ser usado como predictor de vector de movimiento para un bloque que se codifica actualmente. Por ejemplo, si un vector de movimiento de disparidad identifica una imagen de referencia que tiene el mismo identificador de vista que un vector de movimiento actual que se predice, y el vector de movimiento de disparidad tiene una imagen objetivo con el mismo identificador de vista que el vector de movimiento actual que se predice, el vector de movimiento tal vez no se escale antes de ser usado para predecir el vector de movimiento para el bloque actual. En
25 otros casos, el vector de movimiento de disparidad puede escalarse antes de ser usado para predecir el vector de movimiento para el bloque actual.

30 En otro ejemplo, un vector de movimiento de disparidad puede predecirse a partir de un vector de movimiento de disparidad asociado con un bloque contiguo en el espacio. En este ejemplo, si el identificador de vista de la imagen de referencia del vector de movimiento de disparidad es el mismo que el de la imagen de referencia del vector de movimiento que se va a predecir (por ejemplo, el vector de movimiento asociado con el bloque que se predice actualmente), tal vez no se precise escalado. De lo contrario, el vector de movimiento de disparidad se puede escalar basándose en la ubicación de cámara de una cámara usada para captar los datos de vídeo. Es decir, por ejemplo, el vector de movimiento de disparidad que se usa para la predicción puede escalarse de acuerdo con
35 una diferencia entre el identificador de vista de la imagen de referencia del vector de movimiento de disparidad y el identificador de vista de la imagen objetivo del vector de movimiento. En algunos ejemplos, el escalado del vector de movimiento de disparidad puede realizarse basándose en las traslaciones de las vistas.

40 En otro ejemplo, un vector de movimiento de disparidad puede predecirse a partir de un vector de movimiento de disparidad asociado con un bloque contiguo en el tiempo. En este ejemplo, si el identificador de vista de la imagen de referencia del vector de movimiento de disparidad es el mismo que el de la imagen de referencia del vector de movimiento que se va a predecir, y el identificador de vista de la imagen objetivo del vector de movimiento de disparidad es el mismo que el de la imagen de referencia del vector de movimiento que se va a predecir, no se precisará escalado. De lo contrario, el vector de movimiento de disparidad se puede escalar basándose en una
45 diferencia en el identificador de vista, como se describe con respecto al ejemplo anterior.

50 Con respecto a la predicción de vector de movimiento temporal, de acuerdo con unos aspectos de esta divulgación, un vector de movimiento temporal que tiene una imagen objetivo en una primera vista puede usarse para predecir un vector de movimiento temporal que tiene una imagen objetivo en una segunda vista diferente. En algunos ejemplos, el bloque de la imagen objetivo del vector de movimiento temporal que se usa para la predicción puede estar coubicado con el bloque que actualmente se predice en una vista diferente. En otros ejemplos, el bloque de la imagen objetivo del vector de movimiento temporal que se usa para la predicción puede estar desplazado respecto del bloque actual, debido a una disparidad entre las dos vistas.

55 En algunos ejemplos, cuando un vector de movimiento que se predice a partir de una vista diferente es un vector de movimiento temporal, el vector de movimiento puede escalarse basándose en una diferencia en las distancias de recuento de orden de imágenes (POC). Por ejemplo, de acuerdo con unos aspectos de esta divulgación, si una imagen de referencia del vector de movimiento temporal que se usa para la predicción tiene el mismo valor de POC que la imagen de referencia del vector de movimiento actual que se predice, y la imagen objetivo del vector de movimiento temporal usado para la predicción tiene el mismo valor de POC que la imagen de referencia del vector de movimiento actual que se está prediciendo, el vector de movimiento que se está usando para la predicción tal vez no se escale. De lo contrario, sin embargo, el vector de movimiento que se usa para la predicción puede escalarse basándose en una diferencia en el valor de POC entre la imagen de referencia del vector de movimiento que se
60 usada para la predicción y la imagen de referencia del vector de movimiento que se está prediciendo actualmente.

65 De acuerdo con algunos aspectos de esta divulgación, se pueden usar vectores de movimiento temporal y/o de

disparidad desde vistas diferentes como MVP candidatos. Por ejemplo, los vectores de movimiento temporal y/o de disparidad pueden usarse para calcular un MVD para un bloque actual. De acuerdo con otros aspectos de esta divulgación, se pueden usar vectores de movimiento temporal y/o disparidad de diferentes vistas como candidatos de fusión. Por ejemplo, los vectores de movimiento temporal y/o de disparidad pueden heredarse para un bloque actual. En dichos ejemplos, se puede usar un valor de índice para identificar al bloque contiguo del que el bloque de vídeo actual hereda su información de movimiento. En cualquier caso, se puede escalar un vector de movimiento de disparidad y/o temporal de una vista diferente de la que se usada como MVP o candidato de fusión antes de usarlo como MVP o candidato de fusión.

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo que puede usar las técnicas para la predicción del vector de predicción en la codificación multivista. 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 decodificar en un momento posterior mediante un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 a través de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de entre una amplia gama de dispositivos, que incluye ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, un dispositivo de transmisión de vídeo en continuo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a decodificar, a través del medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para permitir al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real.

Los datos de vídeo codificados pueden modularse de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa 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.

En algunos ejemplos, pueden transmitirse datos codificados desde la interfaz de salida 22 hasta un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados del dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de entre una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco fijo, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil u o cualquier otro medio adecuado de almacenamiento digital, para almacenar datos de vídeo codificados. En un ejemplo adicional, 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 los datos de vídeo almacenados del dispositivo de almacenamiento a través de transmisión en continuo o descarga. El servidor de ficheros puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los ejemplos de servidores de archivos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, unos dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión wifi), una conexión 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 en continuo, una transmisión de descarga o una combinación de ambas.

Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, como apoyo de cualquiera de entre una diversidad de aplicaciones multimedia, tales como radiodifusiones de televisión, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo de Internet, tales como la transmisión en continuo adaptativa y dinámica a través de HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para dar soporte a la transmisión de vídeo unidireccional o bidireccional, y ofrecer aplicaciones tales como la transmisión de vídeo en continuo, la

reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. De acuerdo con esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede estar configurado para aplicar las técnicas para la predicción del vector de movimiento en la codificación multivista. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde una fuente de vídeo externa 18, tal como una cámara externa. Asimismo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema ilustrado 10 de la figura 1 es simplemente un ejemplo. Las técnicas para la predicción del vector de movimiento en la codificación multivista pueden ser realizadas por cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, en general, las técnicas de esta divulgación se llevan a cabo mediante un dispositivo de codificación de vídeo, las técnicas también pueden llevarse a cabo mediante un codificador/decodificador de vídeo, denominado típicamente "CÓDEC". Además, las técnicas de esta divulgación también pueden llevarse a cabo mediante un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación, donde el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera esencialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y de decodificación de vídeo. Por lo tanto, el sistema 10 puede dar soporte a una transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para transmisión de vídeo en continuo, reproducción de vídeo, difusión de vídeo o videotelefonía.

La fuente de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captación de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo captado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo. Como alternativa adicional, la fuente de vídeo 18 puede generar datos basados en gráficos de ordenador como vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, como se ha mencionado anteriormente, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas. En cada caso, el vídeo captado, precaptado o generado por ordenador puede ser codificado por el codificador de vídeo 20. La interfaz de salida 22 puede entonces transmitir la información de vídeo codificada a un medio legible por ordenador 16.

El medio legible por ordenador 16 puede incluir medios transitorios, tales como una difusión inalámbrica o una transmisión de red cableada, 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 otro tipo de 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, mediante transmisión por red. De manera similar, un dispositivo informático de una instalación de producción de un medio, tal como una instalación de estampado 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, puede considerarse que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de varias formas, en varios ejemplos.

La interfaz de entrada 28 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 sintáctica definida por el codificador de vídeo 20, que también es usada por el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o el procesamiento de bloques y otras unidades codificadas, por ejemplo, GOP. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario y puede comprender cualquiera de entre 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.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en fase de elaboración, y pueden ajustarse al modelo de prueba HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o industriales, tales como la norma ITU-T H.264, también denominada MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de codificación de vídeo incluyen las normas MPEG-2 e ITU-T H.263. Aunque no se representa en la FIG. 1, en algunos aspectos, tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar integrados en un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un

flujo de datos común o en flujos de datos diferentes. Si procede, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexado ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

5 La norma ITU-T H.264/MPEG-4 (AVC) fue formulada por el Grupo de expertos en codificación de vídeo de la UIT-T (VCEG), junto con el Grupo de expertos en películas de ISO/IEC (MPEG), como el producto de una asociación colectiva conocida como Equipo mixto de vídeo (JVT). En algunos aspectos, las técnicas descritas en esta divulgación pueden aplicarse a dispositivos que se ajustan en general a la norma H.264. La norma H.264 se describe en la Recomendación ITU-T H.264, Codificación de vídeo avanzada para servicios audiovisuales genéricos, del Grupo de estudio de la ITU-T, con fecha de marzo de 2005, que se puede denominar en el presente documento
10 norma H.264 o especificación H.264 o norma o especificación H.264/AVC. El Equipo mixto de vídeo (JVT) continúa trabajando en las ampliaciones de las normas H.264/MPEG-4 AVC.

15 El equipo JCT-VC está trabajando en la elaboración de la norma HEVC. Las actividades de normalización de la HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a dispositivos existentes de acuerdo, por ejemplo, con la ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación de predicción intra, el HM puede proporcionar hasta treinta y tres modos de codificación de predicción intra.

20 En general, el modelo de funcionamiento del HM indica que una imagen de vídeo (o "trama") puede dividirse en una secuencia de bloques en árbol o unidades de codificación más grandes (LCU), que incluyen muestras tanto de luma como de croma. Los datos sintácticos de un flujo de bits pueden definir un tamaño para la LCU, que es la unidad de codificación más grande en lo que respecta al número de píxeles. Un segmento incluye un cierto número de bloques en árbol consecutivos por orden de codificación. Una imagen puede dividirse en uno o más segmentos. Cada bloque
25 en árbol puede dividirse en unidades de codificación (CU) de acuerdo con un árbol cuaternario. En general, una estructura de datos en árbol cuaternario incluye un nodo por cada CU, donde un nodo raíz corresponde al bloque en árbol. Si una CU se divide en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU.

30 Cada nodo de la estructura de datos en árbol cuaternario puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo del árbol cuaternario puede incluir un indicador de división, que indica si la CU correspondiente al nodo está dividida o no en varias sub-CU. Los elementos sintácticos de una CU pueden definirse de manera recursiva y pueden depender de si la CU está dividida o no en varias sub-CU. Si una CU no se divide más, se denomina CU hoja. En esta divulgación, cuatro sub-CU de una CU hoja también se denominarán CU
35 hoja, aunque no haya una división explícita de la CU hoja original. Por ejemplo, si una CU con un tamaño de 16x16 no se divide más, las cuatro sub-CU de tamaño 8x8 también se denominarán CU hojas aunque la CU de tamaño 16x16 no se haya dividido nunca.

40 Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto porque una CU no dispone de una distinción de tamaño. Por ejemplo, un bloque en árbol puede dividirse en cuatro nodos hijo (también denominados sub-CU), y cada nodo hijo puede a su vez ser un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final, no dividido, denominado nodo hoja del árbol cuaternario, comprende un nodo de codificación, es decir, un nodo denominado también CU hoja. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces en que puede dividirse un bloque en árbol, denominado profundidad máxima de CU, y también
45 pueden definir un tamaño mínimo de los nodos de codificación. Por consiguiente, un flujo de bits también puede definir una unidad que es la unidad de codificación más pequeña (SCU). Esta divulgación usará el término "bloque" para referirse a cualquiera de una CU, PU o TU, en el contexto de la HEVC, o a unas estructuras de datos similares en el contexto de otras normas (por ejemplo, macrobloques y subbloques de los estos en la norma H.264/AVC).

50 Una CU incluye un nodo de codificación y unas unidades de predicción (PU) y unas unidades de transformada (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación, y debe ser de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división
55 pueden diferir dependiendo de si la CU está codificada en modo de salto o directo, codificada en modo de predicción intra o codificada en modo de predicción inter. Las PU pueden dividirse para no tener forma cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

60 La norma HEVC admite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU típicamente se basa en el tamaño de las PU de una CU determinada definida para una LCU dividida, aunque esto puede no ser siempre así. Las TU presentan típicamente el mismo tamaño o un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas mediante una estructura en árbol cuaternario conocida como "árbol cuaternario residual"
65 (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformada (TU). Los valores de diferencias de píxeles asociados a las TU pueden transformarse para generar coeficientes de transformada, que pueden

cuantificarse.

Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa una zona espacial correspondiente a la totalidad o una parte de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. En general, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada en modo intra, pueden incluirse datos para la PU en un árbol cuaternario residual (RQT), que pueden incluir datos que describen un modo de predicción intra para una TU correspondiente a la PU. En otro ejemplo, cuando la PU está codificada en modo inter, la PU puede incluir datos que definen uno o más vectores de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden indicar, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, con una precisión de píxel de un cuarto o una precisión de píxel de un octavo), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, Lista 0, Lista 1 o Lista C) para el vector de movimiento.

Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Las unidades de transformada pueden especificarse mediante un RQT (también denominado estructura en árbol cuaternario de TU), como la descrita anteriormente. Por ejemplo, un indicador de división puede indicar si una CU hoja está dividida en cuatro unidades de transformada. A continuación, cada unidad de transformada puede dividirse en otras sub-TU. Cuando una TU no se divide más, puede denominarse TU hoja. En general, para la codificación intra, todas las TU hoja que pertenecen a una CU hoja comparten el mismo modo de predicción intra. Es decir, el mismo modo de predicción intra se aplica en general para calcular valores predichos para todas las TU de una CU hoja. Para la codificación intra, un codificador de vídeo 20 puede calcular un valor residual para cada TU hoja mediante la modalidad de predicción intra, como una diferencia entre la parte de la CU correspondiente a la TU y el bloque original. Una TU no está necesariamente limitada al tamaño de una PU. De este modo, las TU pueden ser más grandes o pequeñas que una PU. Para la codificación intra, una PU puede estar ubicada con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder con el tamaño de la CU hoja correspondiente.

Además, las TU de las CU hoja también pueden asociarse a unas respectivas estructuras de datos en árbol cuaternario, denominadas árboles cuaternarios residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuaternario que indica cómo la CU hoja está dividida en varias TU. El nodo raíz de un árbol cuaternario de TU corresponde en general a una CU hoja, mientras que el nodo raíz de un árbol cuaternario de CU corresponde en general a un bloque en árbol (o LCU). Las TU del RQT que no están divididas se denominan TU hoja. En general, en esta divulgación se usan los términos CU y TU para hacer referencia a una CU hoja y a una TU hoja, respectivamente, a no ser que se indique lo contrario.

Una secuencia de vídeo incluye típicamente una serie de imágenes. Tal como se describe en el presente documento, "imagen" y "trama" pueden usarse indistintamente. Es decir, una imagen que contiene datos de vídeo se puede denominar trama de vídeo o simplemente "trama". Un grupo de imágenes (GOP) comprende en general una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más de las imágenes o en otras ubicaciones, que indican un número de imágenes incluidas en el GOP. Cada segmento de una imagen puede incluir datos sintácticos de segmento que describen un modo de codificación para el segmento respectivo. Un codificador de vídeo 20 actúa típicamente sobre bloques de vídeo de segmentos de vídeo individuales, con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación de una CU. Los bloques de vídeo pueden presentar tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

En un ejemplo, el HM admite la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular sea $2N \times 2N$, el HM admite la predicción intra en tamaños de PU de $2N \times 2N$ o $N \times N$, y la predicción inter en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la división asimétrica para la predicción inter en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida en dos partes de 25 % y 75 %. La parte de la CU correspondiente a la división de 25 % está indicada por una "n" seguida de una indicación "arriba", "abajo", "izquierda" o "derecha". Así, por ejemplo, " $2N \times nU$ " se refiere a una CU $2N \times 2N$ que está dividida horizontalmente con una PU $2N \times 0,5N$ encima y una PU $2N \times 1,5N$ debajo.

En esta divulgación, " $N \times N$ " y "N por N" pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque de $N \times N$ presenta en general N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque pueden estar dispuestos en filas y columnas. Además, los bloques no deben tener necesariamente el mismo número de píxeles en la dirección horizontal y la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N.

Tras la codificación de predicción intra o predicción inter mediante las PU de una CU, el codificador de vídeo 20

5 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos sintácticos que describen un procedimiento o modo de generación de datos de píxeles predictivos en el dominio espacial (también denominado como el dominio del píxel) y las TU pueden comprender coeficientes en el dominio de la transformada, tras la aplicación de una transformada, por ejemplo, una transformada discreta de coseno (DCT), una transformada entera, una transformada de ondícula o una transformada conceptualmente similar, a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU que incluyen los datos residuales para la CU, y a continuación transformar las TU para generar coeficientes de transformada para la CU.

10 Tras cualquier transformada para generar coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere en general a un proceso en el que los coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o la totalidad de los coeficientes. Por ejemplo, un valor de n bits puede redondearse a la baja hasta un valor de m bits durante la cuantificación, donde n es mayor que m .

20 Después de la cuantificación, el codificador de vídeo puede explorar los coeficientes de transformada, generando un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformada cuantificados. La exploración puede diseñarse para colocar coeficientes de energía más alta (y por lo tanto de menor frecuencia) en la parte delantera de la matriz y para colocar coeficientes de energía más baja (y por lo tanto de mayor frecuencia) en la parte trasera de la matriz. En algunos ejemplos, el codificador de vídeo 20 puede usar un orden de exploración predefinido para explorar los coeficientes de transformada cuantificados y generar un vector en serie que pueda someterse a codificación de entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación de entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa según el contexto (CAVLC), la codificación aritmética binaria adaptativa según el contexto (CABAC), la codificación aritmética binaria adaptativa según el contexto basada en la sintaxis (SBAC), la codificación de entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos de codificación de entropía. El codificador de vídeo 20 también puede realizar la codificación de entropía de elementos sintácticos asociados a los datos de vídeo codificados para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

35 Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto de un modelo contextual a un símbolo que se va a transmitir. El contexto puede referirse, por ejemplo, a si los valores contiguos del símbolo son distintos de cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo que se va a transmitir. Las palabras de código en la VLC pueden construirse de forma que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de la VLC puede permitir un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad puede basarse en un contexto asignado al símbolo.

45 Además, el codificador de vídeo 20 puede enviar datos sintácticos, tales como datos sintácticos basados en bloques, datos sintácticos basados en imágenes y datos sintácticos basados en GOP, al decodificador de vídeo 30, por ejemplo, en una cabecera de imagen, una cabecera de bloque, una cabecera de segmento o una cabecera de GOP. Los datos sintácticos de GOP pueden indicar un cierto número de imágenes del GOP respectivo, y los datos sintácticos de imagen pueden indicar una modalidad de codificación/predicción usada para codificar la imagen correspondiente.

50 El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse, cada uno, como cualquiera de entre una diversidad de circuitos codificadores o decodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de compuertas programables *in situ* (FPGA), circuitos de lógica discreta, software, hardware, firmware o cualquier combinación de estos. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar incluidos cada uno en uno o más codificadores o decodificadores, cada uno de los cuales puede estar integrado como parte de un codificador/decodificador (CÓDEC) de vídeo combinado. Un dispositivo que incluye el codificador de vídeo 20 y/o el decodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

60 La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas descritas en esta divulgación, para predecir vectores de movimiento en la codificación multivista. El codificador de vídeo 20 puede realizar una codificación intra e inter de bloques de vídeo dentro de segmentos de vídeo. La codificación intra se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una imagen determinada. La codificación inter se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo de imágenes adyacentes o de imágenes de una secuencia de vídeo. El modo intra (modo I) puede referirse a cualquiera de varios modos de compresión espacial. Las modalidades inter,

tales como la predicción unidireccional (modalidad P) o la predicción bidireccional (modalidad B), pueden referirse a cualquiera de varias modalidades de compresión basadas en el tiempo.

Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe datos de vídeo por codificar. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de selección de modalidad 40, un sumador 50, una unidad de transformada 52, una unidad de cuantificación 54, una unidad de codificación de entropía 56 y una memoria de imágenes de referencia 64. A su vez, la unidad de selección de modalidad 40 incluye una unidad de estimación de movimiento 42, una unidad de compensación de movimiento 44, una unidad de predicción intra 46 y una unidad de división 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye además una unidad de cuantificación inversa 58, una unidad de transformada inversa 60 y un sumador 62. También puede incluirse un filtro de desbloqueo (no mostrado en la FIG. 2) para filtrar límites de bloque y eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de desbloqueo filtrará típicamente la salida del sumador 62. También pueden usarse filtros de bucle adicionales (de bucle o posbucle), además del filtro de desbloqueo. Dichos filtros no se muestran por razones de brevedad pero, si se desea, pueden filtrar la salida del sumador 50 (como en un filtro de bucle).

Durante el proceso de codificación, el codificador de vídeo 20 recibe una imagen o un segmento que va a codificarse. La imagen o el segmento pueden estar divididos en varios bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 llevan a cabo una codificación de predicción inter del bloque de vídeo recibido con respecto a uno o más bloques de una o más imágenes de referencia para proporcionar compresión temporal. La unidad de predicción intra 46, de forma alternativa, puede llevar a cabo la codificación de predicción intra del bloque de vídeo recibido, con respecto a uno o más bloques contiguos de la misma imagen o segmento que el bloque que va a codificarse, para proporcionar compresión espacial. El codificador de vídeo 20 puede llevar a cabo varias pasadas de codificación, por ejemplo, para seleccionar un modo de codificación adecuada para cada bloque de datos de vídeo.

Además, la unidad de división 48 puede dividir bloques de datos de vídeo en subbloques, basándose en la evaluación de sistemas de división anteriores de pasadas de codificación anteriores. Por ejemplo, la unidad de división 48 puede dividir inicialmente una imagen o un segmento en varias LCU, y dividir cada una de las LCU en varias sub-CU, basándose en un análisis de distorsión de velocidad (por ejemplo, un análisis velocidad-optimización de distorsión). La unidad de selección de modalidad 40 puede generar además una estructura de datos en árbol cuaternario que indica la división de una LCU en varias sub-CU. Las CU de nodos hoja del árbol cuaternario pueden incluir una o más PU y una o más TU.

La unidad de selección de modalidad 40 puede seleccionar una de las modalidades de codificación, intra o inter, por ejemplo, en función de los resultados de error, y proporciona el bloque sometido a codificación intra o inter resultante al sumador 50 para generar datos de bloque residuales, y al sumador 62 para reconstruir el bloque codificado para su uso como imagen de referencia. La unidad de selección de modalidad 40 proporciona además elementos sintácticos, tales como vectores de movimiento, indicadores de modalidad intra, información de división y otro tipo de información sintáctica, a la unidad de codificación de entropía 56.

La unidad de estimación de movimiento 42, la unidad de predicción de vector de movimiento 43 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generar vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo situado dentro de una imagen actual, con respecto a un bloque predictivo situado dentro de una imagen de referencia (u otra unidad codificada), con respecto al bloque actual que se está codificando dentro de la imagen actual (u otra unidad codificada).

Un bloque predictivo es un bloque que se considera que guarda una estrecha coincidencia con el bloque por codificar, en lo que respecta a la diferencia de píxeles, que puede determinarse mediante una suma de diferencia absoluta (SAD), una suma de diferencia de cuadrados (SSD) u otro tipo de métrica de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxel de subentero de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64, que también puede denominarse memoria tampón de imágenes de referencia. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxel fraccionarias de la imagen de referencia. Por tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxel completo y a las posiciones de píxel fraccionario, y generar un vector de movimiento con una precisión de píxel fraccionario.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo de un segmento sometido a codificación inter, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. En general, los datos para un vector de movimiento pueden incluir una lista de imágenes de referencia, un índice en la lista de imágenes de referencia (ref_idx), un componente horizontal y un componente vertical. La imagen de referencia puede seleccionarse entre una primera lista de imágenes de referencia (Lista 0), una segunda lista de imágenes de referencia (Lista 1) o una lista de imágenes de referencia

combinadas (Lista c), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64.

La unidad de estimación de movimiento 42 puede generar y enviar un vector de movimiento que identifica el bloque predictivo de la imagen de referencia a la unidad de codificación de entropía 56 y a la unidad de compensación de movimiento 44. Es decir, la unidad de estimación de movimiento 42 puede generar y enviar datos de vectores de movimiento que identifican la lista de imágenes de referencia que contiene el bloque predictivo, un índice en la lista de imágenes de referencia que identifica la imagen del bloque predictivo y un componente horizontal y vertical para localizar el bloque predictivo dentro de la imagen identificada.

En algunos ejemplos, en lugar de enviar el vector de movimiento real para una PU actual, la unidad de predicción de vector de movimiento 43 puede predecir el vector de movimiento para reducir aún más la cantidad de datos necesarios para transmitir el vector de movimiento. En este caso, en lugar de codificar y transmitir el propio vector de movimiento, la unidad de predicción de vector de movimiento 43 puede generar una diferencia de vector de movimiento (MVD) con respecto a un vector de movimiento conocido (o conocible). El vector de movimiento conocido, que puede usarse con la MVD para definir el vector de movimiento actual, puede definirse mediante un denominado predictor de vector de movimiento (MVP). En general, para ser un MVP válido, el vector de movimiento que se usada para la predicción debe apuntar a la misma imagen de referencia que el vector de movimiento que se está codificando actualmente.

En algunos ejemplos, como los descritos con respecto a la FIG. 5 a continuación, la unidad de predicción de vector de movimiento 43 puede construir una lista de predictores de vector de movimiento candidatos que incluye varios bloques vecinos en direcciones espaciales y/o temporales como candidatos para MVP. De acuerdo con unos aspectos de esta divulgación, descritos con mayor detalle a continuación, los predictores de vectores de movimiento candidatos también pueden identificarse en imágenes de vistas diferentes (por ejemplo, en la codificación multivista). Cuando se dispone de varios predictores de vector de movimiento candidatos (de varios bloques candidatos), la unidad de predicción de vector de movimiento 43 puede determinar un predictor de vector de movimiento para un bloque actual de acuerdo con unos criterios de selección predeterminados. Por ejemplo, la unidad de predicción de vector de movimiento 43 puede seleccionar el predictor más preciso del conjunto de candidatos basándose en el análisis de la velocidad de codificación y la distorsión (por ejemplo, mediante un análisis velocidad-coste de distorsión u otro análisis de eficiencia de codificación). En otros ejemplos, la unidad de predicción de vector de movimiento 43 puede generar un promedio de los predictores de vector de movimiento candidatos. También son posibles otros procedimientos para seleccionar un predictor de vector de movimiento.

Al seleccionar un predictor de vector de movimiento, la unidad de predicción de vector de movimiento 43 puede determinar un índice de predictor de vector de movimiento (*mvp_flag*), que puede usarse para indicar a un decodificador de vídeo (por ejemplo, como el decodificador de vídeo 30) donde puede localizar el MVP en una lista de imágenes de referencia que contiene bloques de MVP candidatos. La unidad de predicción de vector de movimiento 43 puede determinar también la MVD entre el bloque actual y el MVP seleccionado. El índice MVP y la MVD pueden usarse para reconstruir el vector de movimiento.

En algunos ejemplos, la unidad de predicción de vector de movimiento 43 puede, en su lugar, implementar una denominada "modalidad de fusión", en la que la unidad de predicción de vector de movimiento 43 puede "fusionar" información de movimiento (tal como vectores de movimiento, índices de imágenes de referencia, direcciones de predicción u otra información) de un bloque de vídeo predictivo con un bloque de vídeo actual. Por consiguiente, con respecto a la modalidad de fusión, un bloque de vídeo actual hereda la información de movimiento de otro bloque de vídeo conocido (o conocible). La unidad de predicción de vector de movimiento 43 puede construir una lista de candidatos de modalidades de fusión que incluye varios bloques vecinos en direcciones espaciales y/o temporales como candidatos para la modalidad de fusión. La unidad de predicción de vector de movimiento 43 puede determinar un valor de índice (por ejemplo, *merge_idx*), que puede usarse para indicar a un decodificador de vídeo (por ejemplo, como el decodificador de vídeo 30) dónde puede localizar el bloque de vídeo de fusión en una lista de imágenes de referencia que contiene bloques de fusión candidatos.

De acuerdo con unos aspectos de esta divulgación, la unidad de predicción de vector de movimiento 43 puede identificar un predictor de vector de movimiento, por ejemplo, para generar una MVD o para realizar una fusión, en la codificación multivista. Por ejemplo, la unidad de predicción de vector de movimiento 43 puede identificar un vector de movimiento de disparidad a partir de un bloque de un componente de vista diferente al de un bloque actual para predecir el vector de movimiento para el bloque actual. En otros ejemplos, la unidad de predicción de vector de movimiento 43 puede identificar un vector de movimiento temporal a partir de un bloque de un componente de vista diferente al de un bloque actual para predecir el vector de movimiento para el bloque actual.

Con respecto a la predicción de vector de movimiento de disparidad, la unidad de predicción de vector de movimiento 43 puede identificar un vector de movimiento de disparidad candidato de un bloque candidato para predecir un vector de movimiento para un bloque de vídeo que se codifica actualmente (denominado "bloque actual"). El bloque actual puede estar situado en la misma imagen que el bloque candidato (por ejemplo, ser espacialmente contiguo al bloque candidato), o puede estar situado en otra imagen de la misma vista que el bloque

candidato. En algunos ejemplos, la unidad de predicción de vector de movimiento 43 puede identificar un predictor de vector de movimiento que se refiere a una imagen de referencia en una vista diferente a la de un vector de movimiento para el bloque actual. En dichos casos, de acuerdo con las técnicas de esta divulgación, la unidad de predicción de vector de movimiento 43 puede escalar el predictor de vector de movimiento basándose en una diferencia entre las posiciones de cámara de las dos vistas (por ejemplo, la vista a la que se refiere el predictor de vector de movimiento y la vista a la que se refiere el vector de movimiento actual). Por ejemplo, la unidad de predicción de vector de movimiento 43 puede escalar el predictor de vector de movimiento de disparidad de acuerdo con una diferencia entre las dos vistas. En algunos ejemplos, la diferencia entre las dos vistas puede estar representada por una diferencia entre los identificadores de vista (`view_id`) asociados con las vistas.

Con respecto a la predicción de vector de movimiento temporal, la unidad de predicción de vector de movimiento 43 puede identificar un vector de movimiento temporal candidato de un bloque candidato en una vista diferente a la de un bloque actual para predecir un vector de movimiento para el bloque actual. Por ejemplo, la unidad de predicción de vector de movimiento 43 puede identificar un predictor de vector de movimiento temporal candidato en una primera vista que hace referencia a un bloque de una imagen en otra posición temporal de la primera vista. De acuerdo con unos aspectos de esta divulgación, la unidad de predicción de vector de movimiento 43 puede usar el predictor de vector de movimiento temporal candidato identificado para predecir un vector de movimiento asociado con un bloque actual en una segunda vista diferente. El bloque candidato (que incluye el predictor de vector de movimiento candidato) y el bloque actual pueden estar cubricados. Sin embargo, la posición relativa del bloque candidato puede estar desplazada del bloque actual, debido a una disparidad entre las dos vistas.

De acuerdo con unos aspectos de esta divulgación, la unidad de predicción de vector de movimiento 43 puede generar un índice de MVP (`mvp_flag`) y una MVD, o puede generar un índice de fusión (`merge_idx`). Por ejemplo, la unidad de predicción de vector de movimiento 43 puede generar una lista de MVP o candidatos de fusión. De acuerdo con unos aspectos de esta divulgación, los MVP y/o candidatos de fusión incluyen uno o más bloques de video situados en una vista diferente a la de un bloque de video que se está decodificando actualmente.

La compensación de movimiento, llevada a cabo por la unidad de compensación de movimiento 44, puede implicar obtener o generar el bloque predictivo basándose en el vector de movimiento determinado por la unidad de estimación de movimiento 42 y/o la información de la unidad de predicción de vector de movimiento 43. De nuevo, la unidad de estimación de movimiento 42, la unidad de predicción de vector de movimiento 43 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente en algunos ejemplos. Tras recibir el vector de movimiento para la PU del bloque de video actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia.

El sumador 50 forma un bloque de video residual restando unos valores de píxel del bloque predictivo a los valores de píxel del bloque de video actual que se está codificando, y generando valores de diferencia de píxel, como se analiza posteriormente. En general, la unidad de estimación de movimiento 42 lleva a cabo la estimación de movimiento con respecto a los componentes de luma, y la unidad de compensación de movimiento 44 usa vectores de movimiento calculados basándose en los componentes de luma tanto para los componentes de croma como para los componentes de luma. La unidad de selección de modalidad 40 también puede generar elementos sintácticos asociados a los bloques de video y al segmento de video, para su uso por el decodificador de video 30 durante la decodificación de los bloques de video del segmento de video.

La unidad de predicción intra 46 puede realizar la predicción intra de un bloque actual, como alternativa a la predicción inter llevada a cabo por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de predicción intra 46 puede determinar una modalidad de predicción intra para usar en la codificación de un bloque actual. En algunos ejemplos, la unidad de predicción intra 46 puede codificar un bloque actual mediante varias modalidades de predicción intra, por ejemplo, durante diferentes pasadas de codificación, y la unidad de predicción intra 46 (o la unidad de selección de modalidad 40, en algunos ejemplos) puede seleccionar una modalidad adecuada de predicción intra para usar, entre las modalidades probadas.

Por ejemplo, la unidad de predicción intra 46 puede calcular valores de velocidad-distorsión mediante un análisis de velocidad-distorsión para las diversas modalidades de predicción intra probadas, y seleccionar la modalidad de predicción intra que tenga las mejores características de velocidad-distorsión entre las modalidades probadas. El análisis de velocidad-distorsión determina en general una cantidad de distorsión (o de error) entre un bloque codificado y un bloque original, no codificado, que se codificó para generar el bloque codificado, así como una velocidad binaria (es decir, un número de bits) usada para generar el bloque codificado. La unidad de predicción intra 46 puede calcular proporciones a partir de las distorsiones y velocidades para los diversos bloques codificados, a fin de determinar qué modalidad de predicción intra presenta el mejor valor de velocidad-distorsión para el bloque.

Después de seleccionar una modalidad de predicción intra para un bloque, la unidad de predicción intra 46 puede proporcionar información, que indica la modalidad de predicción intra seleccionada para el bloque, a la unidad de codificación de entropía 56. La unidad de codificación de entropía 56 puede codificar la información que indica la modalidad de predicción intra seleccionada. El codificador de video 20 puede incluir datos de configuración en el

flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índices de modalidades de predicción intra y una pluralidad de tablas de índices de modalidades de predicción intra modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones de la modalidad de predicción intra más probable, una tabla de índices de modalidades de predicción intra y una tabla modificada de índices de modalidades de predicción intra para usar en cada uno de los contextos.

El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modalidad 40 del bloque de vídeo original que se está codificando. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformada 52 aplica una transformada, tal como una transformada discreta de coseno (DCT) o una transformada conceptualmente similar, al bloque residual, generando un bloque de vídeo que comprende valores residuales de coeficientes de transformada. La unidad de procesamiento de transformada 52 puede llevar a cabo otras transformadas que son conceptualmente similares a la DCT. También podrían usarse transformadas de ondícula, transformaciones enteras, transformadas de subbanda u otros tipos de transformadas. En cualquier caso, la unidad de procesamiento de transformada 52 aplica la transformada al bloque residual, generando un bloque de coeficientes de transformada residuales. La transformada puede convertir la información residual, desde un dominio de valor de píxel a un dominio de transformada, tal como un dominio de frecuencia.

La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir más la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o la totalidad de los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar a continuación una exploración de la matriz que incluye los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación de entropía 56 puede realizar la exploración.

Tras la cuantificación, la unidad de codificación de entropía 56 realiza la codificación de entropía de los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación de entropía 56 puede llevar a cabo la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto y basada en sintaxis (SBAC), la codificación de entropía por división de intervalos de probabilidad (PIPE) u otra técnica de codificación de entropía. En el caso de la codificación de entropía basada en el contexto, el contexto puede basarse en bloques contiguos. Tras la codificación de entropía realizada por la unidad de codificación de entropía 56, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, el decodificador de vídeo 30) o archivar para su posterior transmisión o recuperación.

La unidad de cuantificación inversa 58 y la unidad de transformada inversa 60 aplican la cuantificación inversa y la transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio del píxel, por ejemplo, para su uso posterior como bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia sumando el bloque residual a un bloque predictivo de una de las imágenes de la memoria de imágenes de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxel de subentero para su uso en la estimación de movimiento. El sumador 62 suma el bloque residual reconstruido al bloque de predicción de movimiento compensado, generado por la unidad de compensación de movimiento 44, para generar un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede ser usado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como bloque de referencia para realizar la codificación inter de un bloque en una imagen posterior.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 que puede implementar las técnicas descritas en esta divulgación para predecir vectores de movimiento en la codificación multivista. En el ejemplo de la FIG. 3, el decodificador de vídeo 30 incluye una unidad de decodificación de entropía 80, una unidad de predicción 81, una unidad de cuantificación inversa 86, una unidad de transformada inversa 88, un sumador 90 y una memoria de imágenes de referencia 92. La unidad de predicción 81 incluye la unidad de compensación de movimiento 82 y la unidad de predicción intra 84.

Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un segmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de decodificación de entropía 80 del decodificador de vídeo 30 realiza la decodificación de entropía del flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos sintácticos. La unidad de decodificación de entropía 80 envía los vectores de movimiento y otros elementos sintácticos a la unidad de predicción 81. El decodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel del segmento de vídeo y/o el nivel del bloque de vídeo.

Por ejemplo, a modo de antecedente, el decodificador de vídeo 30 puede recibir datos de vídeo comprimidos que se han comprimido para su transmisión a través de una red como las llamadas "unidades de capa de abstracción de red" o unidades NAL. Cada unidad NAL puede incluir una cabecera que identifica un tipo de datos almacenados en

la unidad NAL. Hay dos tipos de datos que comúnmente se almacenan en unidades NAL. El primer tipo de datos almacenados en una unidad NAL son datos de capa de codificación de vídeo (VCL), que incluyen los datos de vídeo comprimidos. El segundo tipo de datos almacenados en una unidad NAL se denomina datos no VCL, que incluyen información adicional como conjuntos de parámetros que definen datos de cabecera comunes a un gran número de unidades NAL e información de mejora complementaria (SEI).

Por ejemplo, los conjuntos de parámetros pueden contener la información de cabecera de nivel de secuencia (por ejemplo, en conjuntos de parámetros de secuencia (SPS)) e información de cabecera de nivel de imagen, que cambia con poca frecuencia (por ejemplo, en conjuntos de parámetros de imagen (PPS)). La información que cambia con poca frecuencia contenida en los conjuntos de parámetros no necesita repetirse para cada secuencia o imagen, mejorando así la eficiencia de codificación. Además, el uso de conjuntos de parámetros permite la transmisión fuera de banda de la información de cabecera, evitando de ese modo la necesidad de transmisiones redundantes para recuperarse de los errores.

Cuando el segmento de vídeo se codifica como un segmento de codificación intra (I), la unidad de predicción intra 84 de la unidad de predicción 81 puede generar datos de predicción para un bloque de vídeo del segmento de vídeo actual, basándose en una modalidad de predicción intra indicado y unos datos de bloques previamente decodificados de la imagen actual. Cuando la imagen de vídeo está codificada como un segmento de codificación inter (por ejemplo, B, P o GPB), la unidad de compensación de movimiento 82 de la unidad de predicción 81 genera bloques predictivos para un bloque de vídeo del segmento de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad de decodificación de entropía 80. Los bloques predictivos pueden generarse a partir de una de las imágenes de referencia de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de imágenes de referencia, Lista 0 y Lista 1, mediante técnicas de construcción predeterminadas, basándose en las imágenes de referencia almacenadas en la memoria de imágenes de referencia 92.

La unidad de compensación de movimiento 82 determina la información de predicción para un bloque de vídeo del segmento de vídeo actual, analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad de compensación de movimiento 82 usa algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, predicción intra o predicción inter), usada para codificar los bloques de vídeo del segmento de vídeo, un tipo de segmento de predicción inter (por ejemplo, segmento B, segmento P o segmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de vídeo sometido a codificación inter del segmento, el estado de predicción inter para cada bloque de vídeo sometido a codificación inter del segmento y otra información, para decodificar los bloques de vídeo en el segmento de vídeo actual. En algunos ejemplos, la unidad de compensación de movimiento 82 puede recibir cierta información de movimiento desde la unidad de predicción de vector de movimiento 83.

De acuerdo con unos aspectos de esta divulgación, la unidad de predicción de vector de movimiento 83 puede recibir datos de predicción que indican dónde puede recuperar información de movimiento para un bloque actual. Por ejemplo, la unidad de predicción de vector de movimiento 83 puede recibir información de predicción de vector de movimiento, tal como un índice de MVP (mvp_flag), una MVD, un indicador de fusión (merge_flag) y/o un índice de fusión (merge_idx) y usar dicha información para identificar información de movimiento usada para predecir un bloque actual. Es decir, como se ha indicado anteriormente con respecto al codificador de vídeo 20, de acuerdo con unos aspectos de esta divulgación, la unidad de predicción de vector de movimiento 83 puede recibir un índice de MVP (mvp_flag) y una MVD y usar dicha información para determinar un vector de movimiento usado para predecir un bloque actual. La unidad de predicción de vector de movimiento 83 puede generar una lista de MVP o candidatos de fusión. De acuerdo con unos aspectos de esta divulgación, los MVP y/o candidatos de fusión pueden incluir uno o más bloques de vídeo situados en una vista diferente a la de un bloque de vídeo que se está decodificando actualmente.

La unidad de predicción de vector de movimiento 83 puede usar un MVP o un índice de fusión para identificar la información de movimiento usada para predecir el vector de movimiento de un bloque actual. Es decir, por ejemplo, la unidad de predicción de vector de movimiento 83 puede identificar un MVP de una lista de imágenes de referencia mediante el índice de MVP (mvp_flag). La unidad de predicción de vector de movimiento 83 puede combinar el MVP identificado con una MVD recibida para determinar el vector de movimiento para el bloque actual. En otros ejemplos, la unidad de predicción de vector de movimiento 83 puede identificar un candidato de fusión a partir de una lista de imágenes de referencia mediante un índice de fusión (merge_idx) para determinar la información de movimiento para el bloque actual. En cualquier caso, después de determinar la información de movimiento para el bloque actual, la unidad de predicción de vector de movimiento 83 puede generar el bloque predictivo para el bloque actual.

De acuerdo con unos aspectos de esta divulgación, la unidad de predicción de vector de movimiento 83 puede determinar un predictor de vector de movimiento en la codificación multivista. Por ejemplo, la unidad de predicción de vector de movimiento 83 puede recibir información que especifica un vector de movimiento de disparidad de un bloque en un componente de vista diferente al de un bloque actual que se usa para predecir el vector de movimiento

para el bloque actual. En otros ejemplos, la unidad de predicción de vector de movimiento 83 puede recibir información que identifica un vector de movimiento temporal de un bloque en un componente de vista diferente al de un bloque actual que se usa para predecir el vector de movimiento para el bloque actual.

5 Con respecto a la predicción del vector de movimiento de disparidad, la unidad de predicción de vector de movimiento 83 puede predecir un vector de movimiento de disparidad para el bloque actual a partir de un bloque candidato. El bloque candidato puede estar situado en la misma imagen que el bloque actual (por ejemplo, espacialmente contiguo al bloque candidato), o puede estar situado en otra imagen de la misma vista que el bloque actual. El bloque candidato también puede estar situado en una imagen de una vista diferente, pero en el mismo
10 instante de tiempo que el bloque actual.

Por ejemplo, con respecto al MVP o a la modalidad de fusión, se conocen (se han determinado previamente) la imagen objetivo y la imagen de referencia para un vector de movimiento de disparidad "A" del bloque actual que se va a predecir. Por motivos descriptivos, se va a suponer que el vector de movimiento de un bloque candidato es "B".
15 De acuerdo con los aspectos de esta divulgación, si el vector de movimiento B no es un vector de movimiento de disparidad, la unidad de predicción del vector de movimiento 83 puede considerar que el bloque candidato no está disponible (por ejemplo, no está disponible para predecir el vector de movimiento A). Es decir, la unidad de predicción de vector de movimiento 83 puede inhabilitar la capacidad de usar el bloque candidato para propósitos de predicción del vector de movimiento.

20 Si el vector de movimiento B es un vector de movimiento de disparidad y la imagen de referencia del vector de movimiento B pertenece a la misma vista que la de la imagen de referencia del vector de movimiento de disparidad A, y la imagen objetivo del vector de movimiento B pertenece a la misma vista que la imagen objetivo del vector de movimiento de disparidad A, la unidad de predicción de vector de movimiento 83 puede usar el vector de movimiento B directamente como un predictor candidato del vector de movimiento A. De lo contrario, la unidad de predicción de vector de movimiento 83 puede escalar el vector de movimiento de disparidad B antes de que pueda usarse como predictor candidato del vector de movimiento A. En dichos casos, de acuerdo con las técnicas de esta divulgación, la unidad de predicción de vector de movimiento 83 puede escalar el vector de movimiento de disparidad basándose en una distancia de vista del vector de movimiento A y una distancia de vista del vector de movimiento B. Por
25 ejemplo, la unidad de predicción de vector de movimiento 83 puede escalar el vector de movimiento de disparidad B mediante un factor de escala que es igual a la distancia de vista del vector de movimiento A dividida por la distancia de vista del vector de movimiento B. En algunos ejemplos, la unidad de predicción de vector de movimiento 83 puede realizar dicho escalado mediante los identificadores de vistas de las imágenes de referencia y las imágenes objetivo.

35 Con respecto a la predicción del vector de movimiento temporal, la unidad de predicción de vector de movimiento 83 puede predecir un vector de movimiento temporal para el bloque actual a partir de un bloque candidato en una vista diferente a la del bloque actual. Por ejemplo, la unidad de predicción de vector de movimiento 83 puede identificar un predictor de vector de movimiento temporal candidato que tiene una imagen objetivo en una primera vista y se refiere a un bloque de una imagen de referencia situada en otra posición temporal de la primera vista. De acuerdo con unos aspectos de esta divulgación, la unidad de predicción de vectores de movimiento 83 puede usar el predictor de vector de movimiento temporal candidato identificado para predecir un vector de movimiento asociado con el bloque actual en una segunda vista diferente.

45 Por ejemplo, con respecto a la modalidad de MVP o de fusión, se conoce (se ha determinado previamente) la imagen objetivo y la imagen de referencia para un vector de movimiento temporal "A" del bloque actual que se va a predecir. Con fines descriptivos, se va a suponer que el vector de movimiento de un bloque candidato es "B". De acuerdo con unos aspectos de esta divulgación, si el vector de movimiento B del bloque candidato no es un vector de movimiento temporal, la unidad de predicción de vector de movimiento 83 puede considerar que el bloque candidato no está disponible (por ejemplo, no está disponible para predecir el vector de movimiento A). Es decir, en
50 algunos ejemplos, la unidad de predicción de vector de movimiento 83 puede desactivar la capacidad de usar el bloque candidato para propósitos de predicción de vector de movimiento.

Si el vector de movimiento B es un vector de movimiento temporal y el POC de la imagen de referencia del vector de movimiento B es el mismo que la imagen de referencia del vector de movimiento A y el POC de la imagen objetivo del vector de movimiento B es el mismo que la imagen objetivo del vector de movimiento B, la unidad de predicción de vector de movimiento 83 puede usar el vector de movimiento B directamente como predictor candidato del vector de movimiento A. De lo contrario, la unidad de predicción de vector de movimiento 83 puede escalar el vector de movimiento temporal B basándose en la distancia temporal. El bloque candidato (que incluye el predictor de vector de movimiento candidato) y el bloque actual pueden estar coubicados en una vista diferente. Sin embargo, la posición relativa del bloque candidato puede estar desplazada del bloque actual, debido a una disparidad entre las
55 dos vistas.

65 La unidad de cuantificación inversa 86 realiza la cuantificación inversa, es decir, descuantifica, los coeficientes de transformada cuantificados, proporcionados en el flujo de bits y decodificados por la unidad de decodificación de entropía 80. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado

por el codificador de vídeo 20 para cada bloque de vídeo del segmento de vídeo, a fin de determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse.

La unidad de transformada inversa 88 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada inversa entera o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada con el fin de generar bloques residuales en el dominio del píxel. De acuerdo con los aspectos de esta divulgación, la unidad de transformada inversa 88 puede determinar la manera en la que las transformadas se aplicaron a los datos residuales. Es decir, por ejemplo, la unidad de transformada inversa 88 puede determinar un RQT que represente la manera en la que las transformadas (por ejemplo DCT, transformada entera, transformada de ondícula o una o más de otras transformadas) se aplicaron a las muestras de luma residuales y las muestras de chroma residuales asociadas con un bloque de datos de vídeo recibidos.

Después de que la unidad de compensación de movimiento 82 genere el bloque predictivo para el bloque de vídeo actual, basándose en los vectores de movimiento y a otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales procedentes de la unidad de transformada inversa 88 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 82. El sumador 90 representa el componente o los componentes que llevan a cabo esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques decodificados, con el fin de eliminar distorsiones de efecto pixelado. Otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) también pueden usarse para allanar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo decodificados de una imagen dada se almacenan a continuación en la memoria de imágenes de referencia 92, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La memoria de imágenes de referencia 92 almacena también vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

La FIG. 4 es un diagrama conceptual que ilustra un patrón de predicción de MVC. En el ejemplo de la FIG. 4, se ilustran ocho vistas y se ilustran doce ubicaciones temporales para cada vista. En general, cada fila de la FIG. 4 corresponde a una vista, mientras que cada columna indica una ubicación temporal. Cada una de las vistas se puede identificar mediante un identificador de vista ("view_id"), que puede usarse para indicar una ubicación relativa de la cámara con respecto a las otras vistas. En el ejemplo mostrado en la FIG. 4, los ID de vista indicados son los ID "S0" a "S7", aunque también se pueden usar ID de vista numéricos. Además, cada una de las ubicaciones temporales se puede identificar mediante un valor de recuento de orden de imágenes (POC), que indica un orden de visualización de las imágenes. En el ejemplo mostrado en la FIG. 4, los valores de POC indicados son los valores "T0" a "T11".

Aunque la MVC tiene una denominada vista base que es decodificable por los decodificadores H.264/AVC y la MVC admite los pares de vistas estéreo, la MVC puede admitir más de dos vistas como entrada de vídeo 3D. Por consiguiente, un renderizador de un cliente que tiene un decodificador de MVC puede esperar un contenido de vídeo 3D con varias vistas.

Las imágenes en la FIG. 4 se indican mediante un bloque sombreado que incluye una letra, que indica si la imagen correspondiente se ha sometido a codificación intra (es decir, es una trama I), o a codificación inter en una dirección (es decir, es una trama P) o en varias direcciones (es decir, es una trama B). En general, las predicciones se indican mediante flechas, donde la imagen a la que se apunta usa el objeto desde el que se apunta como referencia de predicción. Por ejemplo, la trama P de la vista S2 en la ubicación temporal T0 se predice a partir de la trama I de la vista S0 en la ubicación temporal T0.

Al igual que con la codificación de vídeo de vista única, las imágenes de una secuencia de vídeo multivista pueden codificarse predictivamente con respecto a las imágenes de diferentes ubicaciones temporales. Por ejemplo, la trama b de la vista S0 en la ubicación temporal T1 tiene una flecha que la está apuntando desde la trama I de la vista S0 en la ubicación temporal T0, lo cual indica que la trama b se predice a partir de la trama I. Además, sin embargo, en el contexto de la codificación de vídeo multivista, las imágenes pueden someterse a predicción entre vistas. Es decir, un componente de vista puede usar los componentes de vista de otras vistas como referencia. En la MVC, por ejemplo, la predicción entre vistas se realiza como si el componente de vista de otra vista fuera una referencia de predicción intra. Las referencias potenciales entre vistas pueden indicarse en la extensión Conjunto de parámetros de secuencia (SPS) de la MVC y pueden modificarse mediante el proceso de creación de una lista de imágenes de referencia, lo cual permite ordenar con flexibilidad las referencias de predicción inter o de predicción entre vistas.

La FIG. 4 proporciona varios ejemplos de la predicción entre vistas. Las imágenes de la vista S1, en el ejemplo de la FIG. 4, ilustradas se han predicho a partir de imágenes de diferentes ubicaciones temporales de la vista S1, y también se han sometido a predicción entre vistas a partir de imágenes de imágenes de las vistas S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la trama b de la vista S1 en la ubicación temporal T1 se predice a partir de cada una de las tramas B de la vista S1 en las ubicaciones temporales T0 y T2, así como las tramas b de las vistas S0 y S2 en la ubicación temporal T1.

En el ejemplo de la FIG. 4, la letra "B" mayúscula y la "b" minúscula pretenden indicar diferentes relaciones jerárquicas entre las imágenes, en lugar de diferentes procedimientos de codificación. En general, las tramas con "B" mayúscula ocupan una posición relativamente más alta en la jerarquía de predicción que las tramas con "b" minúscula. La FIG. 4 también ilustra las variaciones en la jerarquía de predicción mediante diferentes niveles de sombreado, donde las imágenes con un mayor grado de sombreado (es decir, relativamente más oscuras) ocupan una posición más alta en la jerarquía de predicción que las imágenes que tienen menos sombreado (es decir, son relativamente más claras). Por ejemplo, todas las tramas I de la FIG. 4 se ilustran con sombreado completo, mientras que las tramas P tienen un sombreado algo más claro y las tramas B (y las tramas con b minúscula) tienen diversos niveles de sombreado unas en relación con otras, pero que siempre es más claro que el sombreado de las tramas P y las tramas I.

En general, la jerarquía de predicción se relaciona con índices de orden de vistas, de conformidad con lo cual las imágenes que ocupan posiciones relativamente más altas en la jerarquía de predicción deberían decodificarse antes que las imágenes que ocupan posiciones relativamente más bajas en la jerarquía, de tal modo que esas imágenes que ocupan posiciones relativamente más altas en la jerarquía se puedan usar como imágenes de referencia durante la decodificación de las imágenes que ocupan posiciones relativamente más bajas en la jerarquía. Un índice de orden de vistas es un índice que indica el orden de decodificación de los componentes de vista en una unidad de acceso. Los índices de orden de vistas pueden estar implícitos en un conjunto de parámetros, tal como un SPS.

De esta manera, las imágenes usadas como imágenes de referencia pueden decodificarse antes que las imágenes que se codifican con referencia a las imágenes de referencia. Un índice de orden de vistas es un índice que indica el orden de decodificación de los componentes de vista en una unidad de acceso. Para cada índice de orden de vistas *i*, se señala el correspondiente *view_id*. La decodificación de los componentes de vista sigue el orden ascendente de los índices de orden de vistas. Si se presentan todas las vistas, entonces el conjunto de índices de orden de vistas comprende un conjunto ordenado de manera consecutiva de cero a uno inferior al número total de vistas.

En la MVC, puede extraerse un subconjunto de un flujo de bits completo para formar un subflujo de bits que sigue siendo compatible con la MVC. Hay muchos posibles subflujos de bits que las aplicaciones específicas pueden requerir, en función de, por ejemplo, un servicio proporcionado por un servidor, la capacidad, los servicios de apoyo y las capacidades de los decodificadores de uno o más clientes y/o la preferencia de uno o varios clientes. Por ejemplo, un cliente podría requerir solo tres vistas, y podría haber dos escenarios. En un ejemplo, un cliente puede requerir una experiencia de visualización sin problemas y podría preferir vistas con los valores de *view_id* S0, S1 y S2, mientras que otro cliente puede requerir la escalabilidad de las vistas y preferir vistas con los valores de *view_id* S0, S2 y S4. Debe observarse que ambos subflujos de bits se pueden decodificar como flujos de bits independientes de MVC y pueden admitirse de forma simultánea.

La FIG. 5 es un diagrama de bloques que ilustra potenciales predictores de vector de movimiento candidatos cuando se realiza la predicción del vector de movimiento (que incluye la modalidad de fusión). Es decir, para el bloque 100 que se está codificando actualmente, la información de movimiento (por ejemplo, un vector de movimiento que comprende un componente horizontal y un componente vertical, índices del vector de movimiento, direcciones de predicción u otra información) de los bloques vecinos A0, A1, B0, B1 y B2 puede usarse para predecir la información de movimiento para el bloque 100. Además, la información de movimiento asociada con el bloque cúbico COL también puede usarse para predecir la información de movimiento para el bloque 100. Los bloques vecinos A0, A1, B0, B1 y B2 y el bloque cúbico COL, en el contexto de la predicción del vector de movimiento, pueden denominarse en general y en lo sucesivo predictores de vector de movimiento candidatos.

En algunos ejemplos, los predictores de vector de movimiento candidatos mostrados en la FIG. 5 pueden identificarse al realizar la predicción del vector de movimiento (por ejemplo, ya sea al generar una MVD o al ejecutar una modalidad de fusión). En otros ejemplos, pueden identificarse diferentes candidatos al realizar la modalidad de fusión y la predicción del vector de movimiento. Es decir, un codificador de vídeo puede identificar un conjunto de candidatos de predictor de vector de movimiento para realizar la modalidad de fusión, que es diferente al usado para realizar la predicción del vector de movimiento.

Para realizar la modalidad de fusión, en un ejemplo, un codificador de vídeo (tal como el codificador de vídeo 20) puede determinar inicialmente qué vectores de movimiento de los predictores de vector de movimiento candidatos están disponibles para fusionarse con el bloque 100. Es decir, en algunos casos, la información de movimiento de uno o más de los predictores de vector de movimiento candidatos pueden no estar disponibles debido, por ejemplo, a que el predictor de vector de movimiento candidato se ha sometido a codificación intra, no está codificado o no existe (por ejemplo, uno o más de los predictores de vector de movimiento candidatos se encuentran en otra imagen o segmento). El codificador de vídeo 20 puede crear una lista de predictores de vector de movimiento candidatos que incluye cada uno de los bloques de predictores de vector de movimiento candidatos disponibles.

Después de construir la lista de candidatos, el codificador de vídeo 20 puede seleccionar un vector de movimiento de la lista de candidatos para usarlo como vector de movimiento para el bloque actual 100. En algunos ejemplos, el codificador de vídeo 20 puede seleccionar el vector de movimiento de la lista de candidatos que mejor coincida con el vector de movimiento para el bloque 100. Es decir, el codificador de vídeo 20 puede seleccionar el vector de

movimiento de la lista de candidatos de acuerdo con un análisis velocidad-distorsión.

El codificador de vídeo 20 puede proporcionar una indicación de que el bloque 100 está codificado mediante la modalidad de fusión. Por ejemplo, el codificador de vídeo 20 puede establecer un indicador u otro elemento sintáctico que indica que el vector de movimiento para el bloque 100 se predice mediante la modalidad de fusión. En un ejemplo, el codificador de vídeo 20 puede indicar que los parámetros de predicción inter para el bloque 100 se infieren de un predictor de vector de movimiento candidato estableciendo `merge_flag [x0][y0]`. En este ejemplo, los índices de matriz `x0`, `y0` pueden especificar la ubicación (`x0`, `y0`) de la muestra de luma superior izquierda del bloque de predicción con respecto a la muestra de luma superior izquierda de la imagen (o segmento).

Además, en algunos ejemplos, el codificador de vídeo 20 puede proporcionar un índice que identifique el candidato de fusión del cual el bloque 100 hereda su vector de movimiento. Por ejemplo, `merge_idx [x0][y0]` puede especificar el índice el candidato de fusión, que identifica una imagen de la lista de candidatos de fusión y donde `x0`, `y0` especifica la ubicación (`x0`, `y0`) de la muestra de luma superior izquierda del bloque de predicción relativo a la muestra de luma superior izquierda de la imagen (o segmento).

Un decodificador de vídeo (tal como el decodificador de vídeo 30) puede realizar etapas similares para identificar el candidato de fusión apropiado cuando decodifica el bloque 100. Por ejemplo, el decodificador de vídeo 30 puede recibir una indicación de que el bloque 100 se predice mediante la modalidad de fusión. En un ejemplo, el decodificador de vídeo 30 puede recibir `merge_flag [x0][y0]`, donde (`x0`, `y0`) especifica la ubicación de la muestra de luma superior izquierda del bloque de predicción con respecto a la muestra de luma superior izquierda de la imagen (o segmento).

Además, el decodificador de vídeo 30 puede crear una lista de candidatos de fusión. Por ejemplo, el decodificador de vídeo 30 puede recibir uno o más elementos sintácticos (por ejemplo, indicadores) que indican bloques de vídeo que están disponibles para la predicción de vector de movimiento. El decodificador de vídeo 30 puede crear una lista de candidatos de fusión basada en los indicadores recibidos. De acuerdo con algunos ejemplos, el decodificador de vídeo 30 puede crear la lista de candidatos de fusión (por ejemplo, `mergeCandList`) de acuerdo con la siguiente secuencia:

1. A_1 , si `availableFlagA1` es igual a 1
2. B_1 , si `availableFlagB1` es igual a 1
3. B_0 , si `availableFlagB0` es igual a 1
4. A_0 , si `availableFlagA0` es igual a 1
5. B_2 , si `availableFlagB2` es igual a 1
6. Col, si `availableFlagCol` es igual a 1

Si varios candidatos de fusión tienen los mismos vectores de movimiento y los mismos índices de referencia, los candidatos de fusión pueden eliminarse de la lista.

El decodificador de vídeo 30 puede identificar el candidato de fusión apropiado de acuerdo con un índice recibido. Por ejemplo, el decodificador de vídeo 30 puede recibir un índice que identifica el candidato de fusión del cual el bloque 100 hereda su vector de movimiento. En un ejemplo, `merge_idx [x0][y0]` puede especificar el índice de candidato de fusión, que identifica una imagen de la lista de candidatos de fusión y donde `x0`, `y0` especifica la ubicación (`x0`, `y0`) de la muestra de luma superior izquierda del bloque de predicción relativa a la muestra de luma superior izquierda de la imagen (o segmento).

En algunos ejemplos, el decodificador de vídeo 30 puede escalar el predictor del vector de movimiento antes de fusionar la información de movimiento del bloque candidato con el bloque 100. Por ejemplo, con respecto a un predictor de vector de movimiento temporal, si el predictor de vector de movimiento se refiere a un bloque predictivo de una imagen de referencia que está situada en una ubicación temporal diferente que el bloque predictivo al que se refiere el bloque 100 (por ejemplo, el vector de movimiento concreto para el bloque 100), el decodificador de vídeo 30 puede escalar el predictor de vector de movimiento. Por ejemplo, el decodificador de vídeo 30 puede escalar el predictor de vector de movimiento de modo que se refiera a la misma imagen de referencia que la imagen de referencia para el bloque 100. En algunos ejemplos, el decodificador de vídeo 30 puede escalar el predictor de vector de movimiento de acuerdo con una diferencia en los valores de recuento de orden de imágenes (POC). Es decir, el decodificador de vídeo 30 puede escalar el predictor de vector de movimiento basándose en una diferencia entre una distancia de POC entre el bloque candidato y el bloque predictivo al que se refiere el predictor de vector de movimiento y una distancia de POC entre el bloque 100 y la imagen de referencia actual (por ejemplo, a la que se refiere el vector de movimiento concreto para el bloque 100). Después de seleccionar el predictor de vector de movimiento apropiado, el decodificador de vídeo 30 puede fusionar la información de movimiento asociada con el predictor de vector de movimiento con la información de movimiento para el bloque 100.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar un proceso similar para realizar la predicción del vector de movimiento para un bloque actual de datos de vídeo. Por ejemplo, el codificador de vídeo 20 puede determinar inicialmente qué vectores de movimiento de los predictores de vector de movimiento candidatos

están disponibles para ser usados como MVP. La información de movimiento de uno o más de los predictores de vector de movimiento candidatos puede no estar disponible debido, por ejemplo, a que el predictor del vector de movimiento candidato se ha sometido a codificación intra, todavía no se ha codificado o no existe.

5 Para determinar cuáles de los predictores de vector de movimiento candidatos están disponibles, el codificador de vídeo 20 puede analizar cada uno de los predictores de vector de movimiento candidatos de uno en uno de acuerdo con un sistema basado en prioridades predeterminado. Por ejemplo, para cada predictor de vector de movimiento candidato, el codificador de vídeo 20 puede determinar si el predictor de vector de movimiento se refiere a la misma imagen de referencia que el vector de movimiento concreto para el bloque 100. Si el predictor de vector de movimiento se refiere a la misma imagen de referencia, el codificador de vídeo 20 puede añadir el predictor de vector de movimiento candidato a una lista de candidatos de MVP. Si el predictor de vector de movimiento no se refiere a la misma imagen de referencia, el predictor de vector de movimiento puede escalarse (por ejemplo, con escalado basado en las distancias de POC, tal como se ha analizado anteriormente) antes de ser añadido a la lista de candidatos de MVP.

15 Con respecto al bloque coubicado COL, si el bloque coubicado incluye más de un predictor de vector de movimiento (por ejemplo, COL se predice como trama B), el codificador de vídeo 20 puede seleccionar uno de los predictores vectoriales de movimiento temporal de acuerdo con la lista actual y la imagen de referencia actual (para el bloque 100). El codificador de vídeo 20 puede entonces añadir el predictor de vector de movimiento temporal seleccionado a la lista de predictores de vector de movimiento candidatos.

20 El codificador de vídeo 20 puede indicar que uno o más predictores de vector de movimiento están disponibles estableciendo `enable_temporal_mvp_flag`. Después de crear la lista de candidatos, el codificador de vídeo 20 puede seleccionar un vector de movimiento de los candidatos para usarlo como predictor de vector de movimiento para el bloque 100. En algunos ejemplos, el codificador de vídeo 20 puede seleccionar el vector de movimiento candidato de acuerdo con un análisis velocidad-distorsión.

25 El codificador de vídeo 20 puede indicar el predictor de vector de movimiento seleccionado mediante un índice de MVP (`mvp_flag`) que identifica el MVP en la lista de candidatos. Por ejemplo, el codificador de vídeo 20 puede establecer `mvp_10_flag [x0][y0]` para especificar el índice de predictor de vector de movimiento de la lista 0, donde `x0`, `y0` especifican la ubicación (`x0`, `y0`) de la muestra de luma superior izquierda del bloque candidato con referencia a la muestra de luma superior izquierda de la imagen. En otro ejemplo, el codificador de vídeo 20 puede establecer `mvp_11_flag [x0][y0]` para especificar el índice de predictor de vector de movimiento de la lista 1, donde `x0`, `y0` especifican la ubicación (`x0`, `y0`) de la muestra de luma superior izquierda del bloque candidato con referencia a la muestra de luma superior izquierda de la imagen. En otro ejemplo más, el codificador de vídeo 20 puede establecer `mvp_1c_flag [x0][y0]` para especificar el índice de predictor de vector de movimiento de la lista `c`, donde `x0`, `y0` especifican la ubicación (`x0`, `y0`) de la muestra de luma superior izquierda del bloque candidato Bloque con referencia a la muestra de luma superior izquierda de la imagen.

30 El codificador de vídeo 20 también puede generar un valor de diferencia de vector de movimiento (MVD). La MVD puede constituir la diferencia entre el predictor de vector de movimiento seleccionado y el vector de movimiento concreto para el bloque 100. El codificador de vídeo 20 puede indicar la MVD con el índice de MVP.

35 El decodificador de vídeo 30 puede realizar operaciones similares para determinar un vector de movimiento para un bloque actual mediante un predictor de vector de movimiento. Por ejemplo, el decodificador de vídeo 30 puede recibir una indicación en un conjunto de parámetros (por ejemplo, un conjunto de parámetros de imagen (PPS)) que indica que la predicción de vector de movimiento está habilitada para una o más imágenes. Es decir, en un ejemplo, el decodificador de vídeo 30 puede recibir `enable_temporal_mvp_flag` en un PPS. Cuando una imagen concreta hace referencia a un PPS en el que `enable_temporal_mvp_flag` es igual a cero, las imágenes de referencia de la memoria de imágenes de referencia pueden marcarse como "no usadas para la predicción de vector de movimiento temporal".

40 Si se implementa la predicción de vector de movimiento, al recibir el bloque 100, el decodificador de vídeo 30 puede crear una lista de candidatos de MVP. El decodificador de vídeo 30 puede usar el mismo sistema analizado anteriormente con respecto al codificador de vídeo 20 para crear la lista de candidatos de MVP. En algunos casos, el decodificador de vídeo 30 en general puede realizar un escalado de vector de movimiento similar al descrito anteriormente con respecto al codificador de vídeo 20. Por ejemplo, si un predictor de vector de movimiento no se refiere a la misma imagen de referencia que el bloque 100, el predictor de vector de movimiento puede escalarse (por ejemplo, mediante escalado basado en las distancias de POC, tal como se ha analizado anteriormente) antes de añadirse a la lista de candidatos de MVP. El decodificador de vídeo 30 puede identificar el predictor de vector de movimiento apropiado para el bloque 100 mediante un índice de MVP recibido (`mvp_flag`) que identifica el MVP en la lista de candidatos. El decodificador de vídeo 30 puede entonces generar el vector de movimiento para el bloque 100 mediante el MVP y una MVD recibida.

45 La FIG. 5 ilustra en general la modalidad de fusión y la predicción de vector de movimiento en una vista única. Debe entenderse que los bloques de predictores de vector de movimiento candidatos mostrados en la FIG. 5 se

proporcionan con fines ejemplificativos solamente, y que se pueden usar más, menos o diferentes bloques con el propósito de predecir información de movimiento. De acuerdo con unos aspectos de esta divulgación, descritos a continuación, la modalidad de fusión y la predicción de vector de movimiento también se pueden aplicar cuando se codifica más de una vista (como en la MVC). En dichos casos, los predictores de vector de movimiento y los bloques predictivos pueden estar localizados en vistas diferentes a las del bloque 100.

La FIG. 6 es un diagrama conceptual que ilustra la generación y el escalado de un predictor de vector de movimiento en la codificación multivista. Por ejemplo, de acuerdo con unos aspectos de esta divulgación, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede escalar un vector de movimiento de disparidad 120 (mv) a partir de un bloque de predictores de vector de movimiento de disparidad candidatos 122 ("bloque candidato") para generar un predictor de vector de movimiento 124 (mv') para el bloque actual 126. Aunque la FIG. 6 se describe con respecto al decodificador de vídeo 30, debe entenderse que las técnicas de esta divulgación pueden llevarse a cabo mediante una variedad de codificadores de vídeo diferentes, incluidos otros procesadores, unidades de procesamiento, unidades de codificación basadas en hardware, tales como codificadores/decodificadores (CÓDEC), y similares.

En el ejemplo de la FIG. 6, el bloque candidato 122 es espacialmente contiguo al bloque actual 126 en el componente de vista dos (view_id 2). El bloque candidato 122 se ha sometido a predicción inter e incluye el vector de movimiento 120 que hace referencia (o "apunta") a un bloque predictivo del componente de vista cero (view_id 0). Por ejemplo, el vector de movimiento 120 tiene una imagen objetivo en la vista dos (view_id 2) y una imagen de referencia en la vista cero (view_id 0). El bloque actual 126 también se ha sometido a predicción inter e incluye un vector de movimiento concreto (no mostrado) que se refiere a un bloque predictivo del componente de vista uno (view_id 1). Es decir, por ejemplo, el vector de movimiento concreto para el bloque actual 126 tiene una imagen objetivo en la vista dos (view_id 2) y un bloque de referencia en la vista uno (view_id 1).

De acuerdo con unos aspectos de esta divulgación, el decodificador de vídeo 30 puede generar un predictor de vector de movimiento 124 para el bloque actual 126 mediante una versión escalada del vector de movimiento 120. Por ejemplo, el decodificador de vídeo 30 puede escalar el vector de movimiento 120 basándose en una diferencia entre las distancias de vista del vector de movimiento 120 y el vector de movimiento concreto para el bloque actual 126. Es decir, el decodificador de vídeo 30 puede escalar el vector de movimiento 120 basándose en una diferencia entre la posición de cámara de una cámara usada para captar el bloque predictivo (en la imagen de referencia) para el bloque candidato 122 y el bloque predictivo (en la imagen de referencia) para el bloque actual 126. En consecuencia, el decodificador de vídeo 30 puede escalar el vector de movimiento de disparidad 120 (por ejemplo, el vector de movimiento que se usa para la predicción) de acuerdo con una diferencia entre el componente de vista al que se refiere el vector de movimiento 120 para el bloque candidato 122 y el componente de vista al que se refiere el vector de movimiento concreto para el bloque actual 126.

En un ejemplo, el decodificador de vídeo 30 puede generar un predictor de vector de movimiento escalado para un bloque actual de acuerdo con la ecuación (1) mostrada a continuación:

$$mv' = mv \left(\frac{ViewDistance(mv')}{ViewDistance(mv)} \right) \quad (1)$$

donde *ViewDistance* (mv) es igual a una diferencia entre un ID de vista de una imagen de referencia del vector de movimiento 120 (por ejemplo, *ViewId*(RefPic(mv)) y un ID de vista de una imagen objetivo del vector de movimiento 120 (por ejemplo, *ViewId*(TargetPic(mv)) y *ViewDistance*(mv') es igual a una diferencia entre un ID de vista de una imagen de referencia del predictor de vector de movimiento 124 (por ejemplo, *ViewId*(RefPic(mv')) y un ID de vista de una imagen objetivo de un predictor de vector de movimiento 124 (por ejemplo, *ViewId*(TargetPic(mv')). Por consiguiente, en este ejemplo, la imagen de referencia del predictor de vector de movimiento 124, RefPic(mv'), pertenece a la nueva vista objetivo y la imagen objetivo del predictor de vector de movimiento 124, TargetPic(mv'), pertenece a la vista actual. De forma similar, la imagen de referencia del vector de movimiento 120, RefPic(mv), pertenece a la vista a la que apunta el vector de movimiento candidato, y la imagen objetivo del vector de movimiento 120, TargetPic(mv), pertenece a la vista actual. Por consiguiente, el decodificador de vídeo 30 puede generar un predictor de vector de movimiento escalado de acuerdo con la ecuación (2) siguiente:

$$mv' = mv \left(\frac{ViewID(NewTarget) - ViewID(Current)}{ViewID(Candidate) - ViewID(Current)} \right) \quad (2)$$

donde mv' representa el predictor de vector de movimiento escalado para el bloque actual, mv representa el vector de movimiento para el bloque candidato, *ViewID*(NewTarget) es el componente de vista al que se refiere el vector de movimiento concreto para el bloque actual, *ViewID*(Current) es el componente de vista del bloque actual y *ViewID*(Candidate) es el componente de vista del bloque candidato.

Aplicando la ecuación (2) al ejemplo de la FIG. 6, mv' representa el predictor de vector de movimiento escalado para el bloque actual 126, mv representa el vector de movimiento 120, *ViewID*(NewTarget) es el componente de vista al

cual se refiere el vector de movimiento 124, $ViewID(Current)$ es el componente de vista del bloque actual 126 y $ViewID(Candidate)$ es el componente de vista del bloque candidato 122. En consecuencia, en el ejemplo mostrado en la FIG. 4, el predictor de vector de movimiento 124 es el vector de movimiento 120

escalado por un factor de la mitad (por ejemplo, $mv' = mv \left(\frac{1-2}{0-2} \right)$). Es decir, el decodificador de vídeo 30 puede

5 escalar tanto el componente de desplazamiento horizontal como el componente de desplazamiento vertical del vector de movimiento 120 por un factor de una mitad para generar el predictor de vector de movimiento 124 para el bloque actual 126.

10 El escalado del vector de movimiento descrito con respecto a la FIG. 6 puede realizarse tanto para la fusión como para la predicción del vector de movimiento. Es decir, por ejemplo, el decodificador de vídeo 30 puede escalar el vector de movimiento 120 antes de fusionar el vector de movimiento 120 con la información de movimiento para el bloque actual 126. En otro ejemplo, el decodificador de vídeo 30 puede escalar el vector de movimiento 120 antes de calcular un valor de diferencia de vector de movimiento (MVD) de acuerdo con una diferencia entre el predictor de vector de movimiento 124 y el vector de movimiento concreto para el bloque actual 126.

15 Como se ilustra en el ejemplo de la FIG. 6, el bloque candidato 122 y el bloque actual 126 pueden estar situados en el mismo componente de vista. Sin embargo, en otros ejemplos, como se describe con mayor detalle respecto a las FIGS. 7 y 8, el bloque candidato puede estar situado en un componente de vista diferente que el bloque actual.

20 La FIG. 7 es otro diagrama conceptual que ilustra la generación y el escalado de un predictor de vector de movimiento. Por ejemplo, de acuerdo con unos aspectos de esta divulgación, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede escalar un vector de movimiento de disparidad 130 (mv) a partir de un bloque de predictor de vector de movimiento de disparidad candidato 132 (x' , y'), para generar el predictor de vector de movimiento 134 (mv') para un bloque actual 136 (x , y), en el que el bloque candidato 132 pertenece a un componente de vista diferente al del bloque actual 136. En consecuencia, el proceso mostrado y descrito con respecto a la FIG. 7 puede denominarse en general proceso de predicción de vector de movimiento de disparidad entre vistas. Aunque la FIG. 7 se describe con respecto al decodificador de vídeo 30, debe entenderse que las técnicas de esta divulgación pueden llevarse a cabo mediante una diversidad de codificadores de vídeo diferentes, incluidos otros procesadores, unidades de procesamiento, unidades de codificación basadas en hardware, tales como codificadores/decodificadores (CÓDEC), y similares.

En el ejemplo mostrado en la FIG. 7, el bloque candidato 132 está situado en el componente de vista uno ($view_id$ 1). El bloque candidato 132 se ha sometido a predicción inter e incluye el vector de movimiento 130 (mv) que se refiere a un bloque predictivo en el componente de vista cero ($view_id$ 0). Por ejemplo, el vector de movimiento 130 tiene una imagen objetivo en la vista uno ($view_id$ 1) y una imagen de referencia en la vista cero ($view_id$ 0). El bloque actual 136 está coubicado con el bloque candidato 132 y situado en el componente de vista dos ($view_id$ 2). Como se describe con más detalle a continuación, en algunos ejemplos, el bloque actual 136 puede incluir un vector de movimiento concreto (no mostrado) que identifica un bloque en una primera vista de referencia ($view_id$ 1). Es decir, por ejemplo, el vector de movimiento concreto para el bloque actual 136 tiene una imagen objetivo en la vista dos ($view_id$ 2) y puede tener un bloque de referencia en la vista uno ($view_id$ 1). En otros ejemplos, el bloque actual puede incluir un vector de movimiento concreto que identifica un bloque en la segunda vista de referencia ($view_id$ 0). Es decir, por ejemplo, el vector de movimiento concreto para el bloque actual 136 tiene una imagen objetivo en la vista dos ($view_id$ 2) y puede tener un bloque de referencia en la vista cero ($view_id$ 0). Por consiguiente, el predictor de vector de movimiento 134 (mv') puede referirse a un bloque de una primera vista de referencia ($view_id$ 1). En otro ejemplo, un segundo predictor de vector de movimiento 138 (mv'') puede referirse a un bloque de una segunda vista de referencia ($view_id$ 0).

En algunos ejemplos, el segundo predictor de vector de movimiento 138 puede no estar disponible para fines de predicción de vector de movimiento. Por ejemplo, el segundo predictor 138 de vector de movimiento solo puede generarse si está disponible un bloque predictivo en la segunda vista de referencia para la predicción directa entre vistas. La disponibilidad de un bloque predictivo en la segunda vista de referencia puede especificarse, por ejemplo, en un conjunto de parámetros (tal como un conjunto de parámetros de secuencia (SPS) o un conjunto de parámetros de imagen (PPS)) o una cabecera de segmento asociada con el bloque actual 136.

De acuerdo con los aspectos de esta divulgación, el decodificador de vídeo puede realizar una predicción de vector de movimiento de disparidad entre vistas mediante la modalidad de fusión o mediante la predicción de vector de movimiento. Con respecto a la modalidad de fusión, el decodificador de vídeo 30 puede seleccionar inicialmente una "vista objetivo" para el bloque 136 actual. En general, la vista objetivo incluye el bloque predictivo para el bloque actual 136. En algunos ejemplos, la vista objetivo puede ser la primera vista de referencia (mostrada en la FIG. 7 como $view_id$ 1). En otros ejemplos, la vista objetivo puede ser la segunda vista de referencia (mostrada en la FIG. 7 como $view_id$ 0). Sin embargo, como se ha indicado anteriormente, en algunos ejemplos la segunda vista de referencia solo puede usarse como vista objetivo si un bloque predictivo de la segunda vista de referencia está disponible para ser usado con fines de predicción entre vistas.

En algunos ejemplos, el decodificador de vídeo 30 puede seleccionar la primera vista de referencia como vista

objetivo. En otros ejemplos, el decodificador de vídeo 30 puede seleccionar, cuando está disponible, la segunda vista de referencia como vista objetivo. La selección de la vista objetivo se puede determinar, por ejemplo, basándose en la disponibilidad de un bloque predictivo y/o un algoritmo de selección predeterminado. El índice de referencia (*ref_idx*) del bloque actual 136 corresponde al índice de la imagen que contiene el bloque predictivo de la vista objetivo, que se añade a la lista de imágenes de referencia del bloque actual 136.

Después de seleccionar la vista objetivo, el decodificador de vídeo 30 puede localizar el bloque candidato 132. En un ejemplo con fines ilustrativos, supongamos que la muestra de luma superior izquierda del bloque actual 136 está situada en una imagen (o segmento) en las coordenadas (*x*, *y*). El decodificador de vídeo 30 puede determinar coordenadas cúbicadas en el componente de vista uno para el bloque candidato 132. Además, en algunos ejemplos, el decodificador de vídeo 30 puede ajustar las coordenadas basándose en una disparidad entre el componente de vista del bloque actual 136 (componente de vista dos) y el componente de vista del bloque candidato (componente de vista uno) 132. En consecuencia, el decodificador de vídeo 30 puede determinar las coordenadas para el bloque candidato 132 como (*x'*, *y'*), donde (*x'*, *y'*) = (*x*, *y*) + disparidad. En algunos ejemplos, la disparidad puede incluirse y/o calcularse en un SPS, PPS, cabecera de segmento, sintaxis de CU y/o sintaxis de PU.

Después de localizar el bloque candidato 132, el decodificador de vídeo 30 puede escalar el vector de movimiento 130 para el bloque candidato 132 basándose en una diferencia entre las distancias de vista del vector de movimiento 130 y el vector de movimiento concreto para el bloque actual 136. Es decir, el decodificador de vídeo 30 puede escalar el vector de movimiento 130 basándose en una diferencia en la posición de cámara de una cámara usada para captar el bloque predictivo para el bloque candidato 132 y el bloque predictivo para el bloque actual 136 (por ejemplo, el bloque predictivo de la vista objetivo). Es decir, el decodificador de vídeo 30 puede escalar el vector de movimiento de disparidad 130 (por ejemplo, el vector de movimiento que se usa para la predicción) de acuerdo con una diferencia entre el componente de vista al que se refiere el vector de movimiento 130 para el bloque candidato 132 y el componente de vista de la vista objetivo.

En un ejemplo, el decodificador de vídeo 30 puede generar un predictor de vector de movimiento escalado para un bloque actual de acuerdo con la ecuación (3) mostrada a continuación:

$$mv' = mv \left(\frac{ViewID(Target) - ViewID(Current)}{ViewID(SecondReference) - ViewID(Reference)} \right) \quad (3)$$

Donde *mv'* representa el predictor de vector de movimiento escalado para el bloque actual, *mv* representa el vector de movimiento para el bloque candidato, *ViewID(Target)* es el componente de vista de la vista objetivo seleccionada, *ViewID(Current)* es el componente de vista del bloque actual, *ViewID(SecondReference)* es el componente de vista de la segunda vista de referencia (si está disponible) y *ViewID(Reference)* es el componente de vista de la primera vista de referencia. En algunos ejemplos, *ViewID(Target)* menos *ViewID(Current)* puede denominarse distancia de vista de predictor de vector de movimiento 134, mientras que *ViewID(SecondReference)* menos *ViewID(Reference)* puede denominarse distancia de vista de vector de movimiento 130. Es decir, la distancia de vista del predictor de vector de movimiento 134 es la diferencia entre la imagen objetivo (*view_id* 1) y la imagen de referencia (*view_id* 2) del predictor de vector de movimiento 134, mientras que la distancia de vista del vector de movimiento 130 es la diferencia entre la imagen objetivo (*view_id* 0) y la imagen de referencia (*view_id* 1) del vector de movimiento 130.

Aplicando la ecuación (3) al ejemplo de la FIG. 7, *mv'* representa el predictor de vector de movimiento escalado 134 o el predictor de vector de movimiento escalado 138, dependiendo de qué componente de vista se selecciona para la vista objetivo. Por ejemplo, si se selecciona la primera vista de referencia (*view_id* 1) como vista objetivo, *mv'* representa el predictor de vector de movimiento escalado 134, *mv* representa el vector de movimiento 130, *ViewID(Target)* es el componente de vista al que se refiere el predictor de vector de movimiento 134, *ViewID(Current)* es el componente de vista del bloque actual 136, *ViewID(SecondReference)* es el componente de vista de la segunda vista de referencia (*view_id* 0) y *ViewID(Reference)* es el componente de vista de la primera vista de referencia (*view_id* 1). En consecuencia, en el ejemplo mostrado en la FIG. 7, el predictor de vector de movimiento 134 es el vector de movimiento 130 escalado por un factor de uno (por ejemplo, $mv' = mv \left(\frac{1-2}{0-1} \right)$).

Es decir, el componente de desplazamiento horizontal y el componente de desplazamiento vertical del vector de movimiento 130 pueden ser iguales al componente de desplazamiento horizontal y el componente de desplazamiento vertical del predictor de vector de movimiento 134.

De forma alternativa, si se selecciona la segunda vista de referencia (*view_id* 0) para la vista de destino, *mv'* representa el predictor de vector de movimiento escalado 138, *mv* representa el vector de movimiento 130, *ViewID(Target)* es el componente de vista al que hace referencia el predictor de vector de movimiento 138, *ViewID(Current)* es el componente de vista del bloque actual 136, *ViewID(SecondReference)* es el componente de vista de la segunda vista de referencia (*view_id* 0) y *ViewID(Reference)* es el componente de vista de la primera vista de referencia (*view_id* 1). En consecuencia, en el ejemplo mostrado en la FIG. 7, el predictor de vector de movimiento 138 es el vector de movimiento 130 escalado por un factor de dos (por ejemplo, $mv' = mv \left(\frac{1-2}{0-1} \right)$).

Es decir, el decodificador de vídeo 30 puede escalar tanto el componente de desplazamiento horizontal como el componente de desplazamiento vertical del vector de movimiento 130 por un factor de dos para generar el predictor de vector de movimiento 138 para el bloque actual 136.

5 De acuerdo con los aspectos de esta divulgación, el decodificador de vídeo 30 puede emprender etapas similares al realizar la predicción del vector de movimiento (por ejemplo, generar un MVP). Por ejemplo, el decodificador de vídeo 30 puede seleccionar una vista objetivo, que puede ser la primera vista de referencia (view_id 1) o la segunda vista de referencia (view_id 0). Sin embargo, si una imagen de referencia del componente de vista que contiene un bloque predictivo para el bloque actual no está disponible para fines de predicción entre vistas, el predictor correspondiente no se puede usar. Por consiguiente, la selección de la vista objetivo puede determinarse, por ejemplo, basándose en la disponibilidad de un bloque predictivo y/o un algoritmo de selección predeterminado.

10 Si no hay ningún bloque predictivo para el bloque actual 136 disponible para su uso en la predicción directa entre vistas en la primera vista de referencia (view_id 1) o la segunda vista de referencia (view_id 0), el decodificador de vídeo 30 no puede realizar la predicción del vector de movimiento. Si hay al menos un bloque predictivo disponible, el decodificador de vídeo 30 puede seleccionar la vista de referencia que incluye el bloque predictivo asociado con el vector de movimiento concreto para el bloque 136 actual.

15 Después de seleccionar una vista objetivo, el decodificador de vídeo 30 puede repetir las etapas descritas anteriormente con respecto a la modalidad de fusión. Por ejemplo, el decodificador de vídeo 30 puede localizar el bloque candidato 132. Es decir, el decodificador de vídeo 30 puede determinar coordenadas cúbicas en el componente de vista uno para el bloque candidato 132. Además, en algunos ejemplos, el decodificador de vídeo 30 puede ajustar las coordenadas basándose en una disparidad entre el componente de vista del bloque actual 136 (componente de vista dos) y el componente de vista del bloque candidato (componente de vista uno) 132.

20 Además, después de localizar el bloque candidato 132, el decodificador de vídeo 30 puede escalar el vector de movimiento 130 para el bloque candidato 132 basándose en una diferencia en la posición de cámara de una cámara usada para captar el bloque predictivo para el bloque candidato 132 y el bloque predictivo para el bloque actual 136 (por ejemplo, el bloque predictivo de la vista objetivo). Es decir, el decodificador de vídeo 30 puede escalar el vector de movimiento de disparidad 130 (por ejemplo, el vector de movimiento que se usa para la predicción) de acuerdo con una diferencia entre el componente de vista al que se refiere el vector de movimiento 130 para el bloque candidato 132 y el componente de vista de la vista objetivo. En algunos ejemplos, el decodificador de vídeo 30 puede realizar el escalado del predictor del vector de movimiento mediante la ecuación (2) anterior. En algunos aspectos, descritos anteriormente con respecto a la FIG. 8 siguiente, el escalado del predictor del vector de movimiento puede extenderse a otras vistas.

25 El decodificador de vídeo 30 puede añadir el bloque candidato 132 a una lista candidata cuando se realiza la modalidad de fusión y/o la predicción del vector de movimiento (descritos, por ejemplo, con respecto a la FIG. 5 anterior). De acuerdo con aspectos de esta divulgación, el bloque candidato puede añadirse a la lista de predictores de vector de movimiento candidatos (por ejemplo, para la modalidad de fusión o la predicción del vector de movimiento con un MVP) de una diversidad de maneras. Por ejemplo, el decodificador de vídeo 30 puede crear la lista de candidatos mediante la localización de candidatos de modalidad de fusión de acuerdo con el sistema siguiente:

- 45
1. A_1 , si $availableFlagA_1$ es igual a 1
 2. V , si $availableFlagV$ es igual a 1
 3. B_1 , si $availableFlagB_1$ es igual a 1
 4. B_0 , si $availableFlagB_0$ es igual a 1
 5. A_0 , si $availableFlagA_0$ es igual a 1
 - 50 6. B_2 , si $availableFlagB_2$ es igual a 1
 7. Col, si $availableFlagCol$ es igual a 1

55 donde V representa el bloque candidato 132. En otros ejemplos, el bloque candidato 132 se puede localizar y añadir a la lista de candidatos en cualquier otra posición de la lista de candidatos.

La FIG. 8 es otro diagrama conceptual que ilustra la generación y el escalado de un predictor de vector de movimiento, de acuerdo con unos aspectos de esta divulgación. Por ejemplo, de acuerdo con unos aspectos de esta divulgación, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede escalar un vector de movimiento de disparidad 140 (mv) a partir de un bloque de predictores de vector de movimiento de disparidad candidatos 142 para generar un predictor de vector de movimiento 144 (mv') para un bloque actual 146, en el que el bloque candidato 142 pertenece a un componente de vista diferente al del bloque actual 146. Aunque la figura 8 se describe con respecto al decodificador de vídeo 30, debe entenderse que las técnicas de esta divulgación pueden llevarse a cabo mediante una variedad de codificadores de vídeo diferentes, incluidos otros procesadores, unidades de procesamiento, unidades de codificación basadas en hardware, tales como codificadores/decodificadores CÓDEC, y similares.

El ejemplo mostrado en la FIG. 8 extiende la predicción de vector de movimiento mostrada y descrita con respecto a la FIG. 7 a un entorno que incluya más de tres vistas. Por ejemplo, como se muestra en la FIG. 8, el bloque candidato 142 está situado en el componente de vista dos (view_id 2). El bloque candidato 142 se ha sometido a predicción inter e incluye el vector de movimiento 140 (mv) que se refiere a un bloque predictivo del componente de vista uno (view_id 1). Por ejemplo, el vector de movimiento 140 tiene una imagen objetivo en la vista dos (view_id 2) y una imagen de referencia en la vista uno (view_id 1). El bloque actual 146 está cubricado con el bloque candidato 142 y situado en el componente de vista tres (view_id 3).

De acuerdo con unos aspectos de esta divulgación, el decodificador de vídeo 30 puede seleccionar una vista objetivo para el bloque actual 146 como componente de vista cero (view_id 0). Por ejemplo, la vista objetivo incluye en general el bloque predictivo para el bloque actual. Si la imagen que contiene el bloque predictivo es una imagen de referencia entre vistas y el bloque predictivo para el bloque actual 146 está situado en una tercera vista de referencia (view_id 0), el decodificador de vídeo 30 puede seleccionar la tercera vista de referencia como vista objetivo.

Después de seleccionar la vista objetivo, el decodificador de vídeo 30 puede localizar el bloque candidato 142. Por ejemplo, suponiendo que la muestra de luma superior izquierda del bloque actual 146 está situada en una imagen (o segmento) de coordenadas (x, y) en el componente de vista tres, el decodificador de vídeo 30 puede determinar unas coordenadas cubricadas en el componente de vista dos para el bloque candidato 142. Además, como se ha indicado anteriormente, el decodificador de vídeo 30 puede ajustar las coordenadas basándose en una disparidad entre el componente de vista del bloque actual 146 (componente de vista tres) y el componente de vista del bloque candidato (componente de vista dos) 142.

Después de localizar el bloque candidato 142, el decodificador de vídeo 30 puede escalar el vector de movimiento 140 para el bloque candidato 142 basándose en una diferencia entre las distancias de vista del vector de movimiento 140 y el vector de movimiento concreto para el bloque actual 146. Es decir, el decodificador de vídeo 30 puede escalar el vector de movimiento 130 basándose en una diferencia en la posición de cámara de una cámara usada para captar el bloque predictivo para el bloque candidato 142 y el bloque predictivo para el bloque actual 146 (por ejemplo, el bloque predictivo de la vista objetivo). Es decir, el decodificador de vídeo 30 puede escalar el vector de movimiento de disparidad 140 (por ejemplo, el vector de movimiento que se usa para la predicción) de acuerdo con una diferencia entre el componente de vista al cual se refiere el vector de movimiento 140 para el bloque candidato 142 y el componente de vista de la vista objetivo (view_id 0).

En un ejemplo, el decodificador de vídeo 30 puede generar un predictor de vector de movimiento escalado para un bloque actual de acuerdo con la ecuación (4) mostrada a continuación:

$$mv' = mv \left(\frac{ViewID(Third) - ViewID(Current)}{ViewID(SecondReference) - ViewID(Reference)} \right) \quad (4)$$

donde mv' representa el predictor de vector de movimiento escalado para el bloque actual, mv representa el vector de movimiento para el bloque candidato, *ViewID(Third)* es el componente de vista de la tercera vista de referencia, *ViewID(Current)* es el componente de vista del bloque actual, *ViewID(SecondReference)* es el componente de vista de la segunda vista de referencia (si está disponible) y *ViewID(Reference)* es el componente de vista de la primera vista de referencia. En algunos ejemplos, *ViewID(Third)* menos *ViewID(Current)* puede denominarse distancia de vista del predictor de vector de movimiento 144, mientras que *ViewID(SecondReference)* menos *ViewID(Reference)* puede denominarse distancia de vista del vector de movimiento 140. Es decir, la distancia de vista del predictor de vector de movimiento 144 es la diferencia entre la imagen objetivo (view_id 0) y la imagen de referencia (view_id 3) del predictor de vector de movimiento 144, mientras que la distancia de vista del vector de movimiento 140 es la diferencia entre la imagen objetivo (view_id 1) y la imagen de referencia (view_id 2) del vector de movimiento 140.

Aplicando la ecuación (3) al ejemplo de la FIG. 8, mv' representa el predictor de vector de movimiento escalado 144. Por ejemplo, *ViewID(Third)* es la tercera vista de referencia (view_id 0), mv' representa el predictor de vector de movimiento escalado 144, mv representa el vector de movimiento 140, *ViewID(Current)* es el componente de vista del bloque actual 146, *ViewID(SecondReference)* es el componente de vista de la segunda vista de referencia (view_id 1), y *ViewID(Reference)* es el componente de vista de la primera vista de referencia (view_id 2). En consecuencia, en el ejemplo mostrado en la FIG. 8, el predictor de vector de movimiento 144 es el vector de movimiento 140 escalado por un factor de tres (por ejemplo, $mv' = mv \left(\frac{0-3}{1-2} \right)$). Es decir, el decodificador de vídeo 30 puede escalar el componente de desplazamiento horizontal y el componente de desplazamiento vertical del vector de movimiento 140 por tres para formar el predictor de vector de movimiento 144.

Mientras que las FIGS. 7 a 8 proporcionan ejemplos para la predicción de vector de movimiento de disparidad entre vistas, debe entenderse que dichos ejemplos se proporcionan meramente con fines ilustrativos. Es decir, las técnicas para la predicción de vector de movimiento de disparidad pueden aplicarse a más o menos vistas que las mostradas. De forma adicional o alternativa, las técnicas para la predicción de vector de movimiento de disparidad

pueden aplicarse en circunstancias en las que las vistas tienen diferentes identificadores de vistas.

La FIG. 9 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de información de predicción para un bloque de datos de vídeo. El ejemplo mostrado en la FIG. 9 se describe en general como si estuviera realizado por un codificador de vídeo. Debe entenderse que, en algunos ejemplos, el procedimiento de la FIG. 9 puede llevarse a cabo mediante el codificador de vídeo 20 (FIGS. 1 y 2) o un decodificador de vídeo 30 (FIGS.1 y 3), descritos anteriormente. En otros ejemplos, el procedimiento de la FIG. 9 puede realizarse mediante una diversidad de diferentes procesadores, unidades de procesamiento, unidades de codificación basadas en hardware, tales como codificadores/decodificadores (CÓDEC), y similares.

De acuerdo con el ejemplo de procedimiento mostrado en la FIG. 9, un codificador de vídeo puede identificar un primer bloque de datos de vídeo en una primera vista, donde el primer bloque de datos de vídeo está asociado con un primer vector de movimiento de disparidad (160). Por ejemplo, el vector de movimiento para el primer bloque de datos de vídeo puede ser un vector de movimiento de disparidad que identifica un bloque de referencia en otro componente de vista. El codificador de vídeo puede entonces determinar si un segundo vector de movimiento asociado con un segundo bloque de datos de vídeo es un vector de movimiento de disparidad (162).

Si el segundo vector de movimiento no es un vector de movimiento de disparidad (la rama NO de la etapa 162), el codificador de vídeo puede identificar un predictor de vector de movimiento candidato diferente (164). Es decir, de acuerdo con algunos aspectos de esta divulgación, la capacidad de usar un vector de movimiento de disparidad (por ejemplo, el primer vector de movimiento) para predecir un vector de movimiento temporal (por ejemplo, el segundo vector de movimiento, cuando el segundo vector de movimiento es un vector de movimiento temporal) puede estar inhabilitado. En dichos casos, el codificador de vídeo puede identificar el primer vector de movimiento como no disponible para su uso con fines de predicción del vector de movimiento.

Si el segundo vector de movimiento es un vector de movimiento de disparidad (la rama SÍ de la etapa 162), el codificador de vídeo puede escalar el primer vector de movimiento para generar un predictor de vector de movimiento para el segundo vector de movimiento (166). Por ejemplo, de acuerdo con aspectos de esta divulgación, el codificador de vídeo puede escalar el primer vector de movimiento para generar el predictor de vector de movimiento de disparidad basándose en diferencias en las distancias de vista asociadas con el primer vector de movimiento de disparidad y el segundo vector de movimiento. Es decir, en algunos ejemplos, el codificador de vídeo puede escalar el predictor del vector de movimiento para el segundo bloque basándose en las localizaciones de la cámara. Por ejemplo, el codificador de vídeo puede escalar el segundo vector de movimiento de acuerdo con una diferencia en los identificadores de vista como se muestra y describe con respecto a las FIGS. 6-8.

El codificador de vídeo puede entonces codificar datos de predicción para el segundo bloque mediante el predictor de vector de movimiento escalado (168). Por ejemplo, el codificador de vídeo puede codificar los datos de predicción para el segundo bloque mediante la modalidad de fusión o mediante la predicción del vector de movimiento. Para la modalidad de fusión, el codificador de vídeo puede codificar directamente los datos de predicción para el segundo bloque mediante el segundo predictor de vector de movimiento escalado. Para la predicción del vector de movimiento, el codificador de vídeo puede codificar los datos de predicción para el segundo bloque generando una MVD. La MVD puede incluir la diferencia entre el primer vector de movimiento y el segundo vector de movimiento escalado.

También debe entenderse que las etapas mostradas y descritas con respecto a la FIG. 9 se proporcionan como un simple ejemplo. Es decir, las etapas del procedimiento de la FIG. 9 no tienen necesariamente que realizarse en el orden mostrado en la FIG. 9, pudiéndose realizar un número inferior de etapas adicionales o alternativas.

La FIG. 10 es un diagrama conceptual que ilustra la generación de un predictor de vector de movimiento a partir de un bloque en una vista diferente a la de un bloque actual. Por ejemplo, de acuerdo con unos aspectos de esta divulgación, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede usar un vector de movimiento temporal 180 (mv) de un bloque de predictor de vector de movimiento temporal candidato 182 para generar el predictor de vector de movimiento 184 (mv') para un bloque actual 186, donde el bloque candidato 182 pertenece a un componente de vista diferente al del bloque actual 186. Aunque la figura 10 se describe con respecto al decodificador de vídeo 30, debe entenderse que las técnicas de esta divulgación pueden llevarse a cabo mediante una diversidad de codificadores de vídeo diferentes, incluidos otros procesadores, unidades de procesamiento, unidades de codificación basadas en hardware, tales como codificadores/decodificadores CÓDEC), y similares.

Como se muestra en la FIG. 10, el bloque actual 186 está situado en el componente de vista uno (view_id 1). El bloque candidato 182 está situado en el componente de vista cero (view_id 0). El bloque candidato 182 se predice temporalmente e incluye el vector de movimiento 180 (mv) que se refiere a un bloque predictivo en una localización temporal diferente del mismo componente de vista. En el ejemplo mostrado en la FIG. 10, el vector de movimiento 180 identifica un bloque predictivo en una imagen que tiene un índice de referencia igual a la variable i (ref_idx = i).

Supongamos que la muestra de luma superior izquierda del bloque actual 186 está situada en una imagen (o

segmento) de coordenadas (x, y) . El decodificador de vídeo 30 puede localizar el bloque candidato 182 determinando unas coordenadas cúbicadas en el componente de vista cero para el bloque candidato 182. En algunos ejemplos, el decodificador de vídeo 30 puede ajustar las coordenadas del bloque candidato 182 basándose en una disparidad entre el componente de vista del bloque actual 186 ($view_id$ 1) y el componente de vista del bloque candidato 182 ($view_id$ 0). En consecuencia, el decodificador de vídeo 30 puede determinar las coordenadas para el bloque candidato 182 como (x', y') , donde $(x', y') = (x, y) + \text{disparidad}$. En algunos ejemplos, la disparidad puede incluirse y/o calcularse en un SPS, PPS, cabecera de segmento, sintaxis de CU y/o sintaxis de PU.

De acuerdo con unos aspectos de esta divulgación, el decodificador de vídeo 30 puede volver a correlacionar el índice de referencia del vector de movimiento 180 que se usa con fines de predicción. En general, como se ha indicado anteriormente, los datos para un vector de movimiento incluyen una lista de imágenes de referencia, un índice en la lista de imágenes de referencia (ref_idx), un componente horizontal y un componente vertical. En la HEVC, puede haber dos listas de imágenes de referencia normales (por ejemplo, lista 0 y lista 1) y una lista de imágenes de referencia combinadas (por ejemplo, lista c). Sin pérdida de generalidad, se va a suponer que la lista de imágenes de referencia actual es la lista t (que puede corresponder a cualquiera de entre la lista 0, la lista 1 o la lista c). De acuerdo con el ejemplo mostrado en la FIG. 10, el vector de movimiento 180 para el bloque candidato 182 puede identificar un bloque predictivo en una imagen situada en el componente de vista cero ($view_id$ 0) que tiene un valor de POC de dos y un ref_idx igual a i . De acuerdo con unos aspectos de esta divulgación, el decodificador de vídeo 30 puede identificar un bloque predictivo cúbicado para el bloque actual 186 en el mismo instante de tiempo que el bloque actual 186. Es decir, el bloque predictivo para el bloque candidato 182 y el bloque predictivo para el bloque actual 186 tienen la misma ubicación temporal, pero se encuentran en imágenes de dos vistas diferentes.

En un ejemplo, si el bloque predictivo identificado para el bloque actual 186 corresponde a la j -ésima imagen de referencia en la lista de imágenes de referencia t para la imagen actual, el decodificador de vídeo 30 puede predecir que el índice de referencia (ref_idx) para el bloque actual 186 es j , y el decodificador de vídeo 30 puede establecer el predictor de vector de movimiento 184 en el mismo valor que el vector de movimiento 180. En consecuencia, el decodificador de vídeo 30 vuelve a correlacionar con eficacia el índice de referencia para el bloque actual 186 a partir de los índices ref_idx i a ref_idx j . Es decir, el decodificador de vídeo 30 determina que el predictor de vector de movimiento 184 para el bloque actual 186 tiene la misma lista de imágenes de referencia, componente horizontal y componente vertical que el bloque candidato 182, sin embargo, el predictor de vector de movimiento 184 se refiere a la j -ésima imagen de referencia de la lista de imágenes de referencia, en lugar de la i -ésima imagen de referencia de la lista de imágenes de referencia.

De acuerdo con unos aspectos de esta divulgación, en algunos ejemplos, el decodificador de vídeo puede también escalar el predictor del vector de movimiento 184. Por ejemplo, si la imagen que contiene el bloque predictivo identificado para el bloque actual 186 no está incluida en la lista de imágenes de referencia t , el decodificador de vídeo 30 puede identificar una segunda imagen que ocupa la posición más cercana en la lista de imágenes de referencia t . En algunos ejemplos, si dos imágenes tienen distancias idénticas a la imagen que contiene el bloque predictivo identificado para el bloque actual 186, el decodificador de vídeo 30 puede seleccionar la imagen que está más próxima a la imagen que contiene el bloque actual 186 como segunda imagen. Supongamos con fines de explicación que la imagen identificada tiene un índice de referencia k . En este ejemplo, el decodificador de vídeo 30 puede entonces predecir que el índice de referencia del predictor de vector de movimiento 184 es k , y el decodificador de vídeo 30 puede escalar el predictor de vector de movimiento 184 basándose en una diferencia en el recuento de orden de imágenes (POC). Es decir, el decodificador de vídeo 30 puede escalar el predictor de vector de movimiento 184 basándose en una diferencia entre la distancia entre el bloque actual 186 y la imagen en el índice de referencia j , y el bloque actual 186 y la imagen en el índice de referencia k .

De acuerdo con algunos ejemplos, el decodificador de vídeo 30 puede realizar el mismo proceso cuando se realiza la predicción del vector de movimiento. Sin embargo, después de determinar el predictor de vector de movimiento 184, el decodificador de vídeo 30 puede generar el vector de movimiento para el bloque actual 186 mediante una MVD. La predicción del vector de movimiento puede usar el mismo proceso. En otro ejemplo, con respecto a la predicción del vector de movimiento, si no puede localizarse un bloque predictivo para el bloque actual 186 (identificado como localizado en el índice de referencia j anterior), el decodificador de vídeo 30 no puede realizar la modalidad de fusión o la predicción de vector de movimiento para el bloque actual 186. Es decir, en lugar de escalar el predictor de vector de movimiento 184, el decodificador de vídeo 30 puede considerar que el predictor de vector de movimiento 184 no está disponible.

El decodificador de vídeo 30 puede añadir el bloque candidato 182 a una lista candidata para realizar la modalidad de fusión y/o la predicción del vector de movimiento (descritos, por ejemplo, con respecto a la FIG. 5 anterior). De acuerdo con unos aspectos de esta divulgación, el bloque candidato 182 se puede añadir a la lista de predictores de vector de movimiento candidatos (por ejemplo, para la modalidad de fusión o la predicción del vector de movimiento con un MVP) de una variedad de formas. Por ejemplo, el decodificador de vídeo 30 puede crear la lista de candidatos localizando candidatos de acuerdo con el siguiente sistema:

1. A_1 , si $availableFlagA_1$ es igual a 1
2. V , si $availableFlagV$ es igual a 1
3. B_1 , si $availableFlagB_1$ es igual a 1
4. B_0 , si $availableFlagB_0$ es igual a 1
5. A_0 , si $availableFlagA_0$ es igual a 1
6. B_2 , si $availableFlagB_2$ es igual a 1
7. Col, si $availableFlagCol$ es igual a 1

donde V representa el bloque candidato 182. En otros ejemplos, el bloque candidato 132 se puede localizar y añadir a la lista de candidatos en cualquier otra posición de la lista de candidatos.

La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de procedimiento para generar un predictor de vector de movimiento. El ejemplo mostrado en la FIG. 11 se describe en general como si fuera realizado por un codificador de vídeo. Debe entenderse que, en algunos ejemplos, el procedimiento de la FIG. 11 puede llevarse a cabo mediante el codificador de vídeo 20 (FIGS. 1 y 2) o el decodificador de vídeo 30 (FIGS. 1 y 3), descritos anteriormente. En otros ejemplos, el procedimiento de la FIG. 11 puede realizarse mediante una variedad de diferentes procesadores, unidades de procesamiento, unidades de codificación basadas en hardware, tales como codificadores/decodificadores (CÓDEC), y similares.

De acuerdo con el ejemplo mostrado en la FIG. 11, el codificador de vídeo puede identificar un primer bloque de datos de vídeo en una primera posición temporal de una primera vista, donde el primer bloque está asociado con un primer vector de movimiento temporal (202). De acuerdo con unos aspectos de esta divulgación, cuando un segundo vector de movimiento asociado con un segundo bloque de datos de vídeo es un vector de movimiento temporal y el segundo bloque es de una segunda vista diferente a la del primer bloque (la rama SÍ de la etapa 204), el codificador de vídeo puede determinar un predictor de vector de movimiento basándose en el primer vector de movimiento (206). Es decir, por ejemplo, el codificador de vídeo puede determinar un predictor de vector de movimiento para predecir el segundo vector de movimiento a partir del primer vector de movimiento. El codificador de vídeo también puede codificar datos de predicción para el segundo bloque mediante el predictor de vector de movimiento (208). Por ejemplo, el codificador de vídeo puede usar el predictor de vector de movimiento en una modalidad de fusión o para generar un valor de MVD.

Si el segundo vector de movimiento no es un vector de movimiento temporal y/o el segundo bloque de datos de vídeo no es de una vista diferente a la del primer bloque de datos de vídeo (la rama NO de la etapa 204), el codificador de vídeo puede determinar si el segundo vector de movimiento es un vector de movimiento de disparidad (210). De acuerdo con unos aspectos de esta divulgación, si el segundo vector de movimiento no es un vector de movimiento de disparidad (la rama NO de la etapa 210), el codificador de vídeo puede identificar un predictor de vector de movimiento candidato diferente (212). Es decir, el codificador de vídeo puede, en algunos ejemplos, no usar el primer vector de movimiento para predecir el segundo vector de movimiento.

Si el segundo vector de movimiento es un vector de movimiento de disparidad (la rama SÍ de la etapa 210), el codificador de vídeo puede determinar si la predicción del vector de movimiento de disparidad está inhabilitada (214). Es decir, de acuerdo con algunos aspectos de esta divulgación, la capacidad de usar un vector de movimiento temporal (por ejemplo, el primer vector de movimiento) para predecir un vector de movimiento de disparidad (por ejemplo, el segundo vector de movimiento, cuando el segundo vector de movimiento es un vector de movimiento de disparidad) puede estar inhabilitada. En dichos casos, el codificador de vídeo puede identificar un predictor de vector de movimiento candidato diferente (212) (la rama NO de la etapa 214).

Si el codificador de vídeo determina que la predicción de vector de movimiento de disparidad está habilitada (por ejemplo, o que la capacidad para habilitar/inhabilitar dicha función no está presente), el codificador de vídeo puede determinar un predictor de vector de movimiento para el segundo vector de movimiento, basándose en el primer vector de movimiento 206) (la rama SÍ de la etapa 214). Además, el codificador de vídeo también puede codificar datos de predicción para el segundo bloque mediante el predictor de vector de movimiento (208). Por ejemplo, el codificador de vídeo puede usar el predictor de vector de movimiento en una modalidad de fusión o para generar un valor de MVD.

También debe entenderse que las etapas mostradas y descritas con respecto a la FIG. 11 se proporcionan como un simple ejemplo. Es decir, las etapas del procedimiento de la FIG. 11 no tienen necesariamente que realizarse en el orden mostrado en la FIG. 11, pudiéndose llevar a cabo un número inferior de etapas adicionales o alternativas.

Deberá entenderse que, según el ejemplo, ciertos actos o sucesos de cualquiera de los procedimientos descritos en el presente documento pueden realizarse en una secuencia distinta, pueden añadirse, fusionarse u omitirse por completo (por ejemplo, no todos los actos o sucesos descritos son necesarios para la puesta en práctica del procedimiento). Además, en ciertos ejemplos, los actos o sucesos pueden realizarse simultáneamente, por ejemplo, mediante el procesamiento multicolor, el procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente. Además, aunque ciertos aspectos de esta divulgación se describen como si fueran realizados por un único módulo o unidad con fines de claridad, debería entenderse que las técnicas de esta divulgación pueden

realizarse mediante una combinación de unidades o módulos asociados con un codificador de vídeo.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse en un medio legible por ordenador y transmitirse como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible, tal como unos medios de almacenamiento de datos o unos medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación.

De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) unos medios de almacenamiento tangibles y legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible a los que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para implementar las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo, y no de manera limitativa, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión puede denominarse adecuadamente medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota mediante un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio.

Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se refieren a medios de almacenamiento tangibles no transitorios. El término disco en el presente documento incluye el disco compacto (CD), el disco de láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, de los cuales el disco flexible normalmente reproduce datos magnética mente, mientras que el resto reproducen datos ópticamente mediante láser. Las combinaciones de los anteriores también deberían incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ejecutarse mediante uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores para usos generales, circuitos integrados de aplicación específica (ASIC), formaciones lógicas programables en el terreno (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador" puede referirse en el presente documento 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, las funciones descritas en el presente documento pueden proporcionarse dentro de hardware dedicado y/o módulos de software configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Asimismo, las técnicas podrían implementarse por completo en uno o más circuitos o elementos lógicos.

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

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

REIVINDICACIONES

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

5 identificar (202) un primer bloque de datos de vídeo en una primera posición temporal de una primera vista, en el que el primer bloque de datos de vídeo está asociado con un primer vector de movimiento temporal; determinar (206) un predictor de vector de movimiento para un segundo vector de movimiento asociado con un segundo bloque de datos de vídeo, en el que el segundo vector de movimiento es un vector de movimiento temporal y el segundo bloque es de una segunda vista que es diferente de la
10 primera vista, en el que determinar el predictor del vector de movimiento comprende:

15 usar el primer vector de movimiento como el predictor de vector de movimiento, cuando un valor de recuento de orden de imágenes (POC) de una imagen de referencia del segundo vector de movimiento es igual a un valor de POC de una imagen de referencia del primer vector de movimiento y un valor de POC de una imagen objetivo del segundo vector de movimiento es igual a un valor de POC de una imagen objetivo del primer vector de movimiento; y

20 codificar (208) datos de predicción para el segundo bloque mediante el predictor de vector de movimiento,

25 estando el procedimiento **caracterizado por que** cuando un valor de POC de una imagen de referencia del vector de movimiento para el segundo bloque es diferente de un valor de POC de una imagen de referencia del primer vector de movimiento, escalar el primer vector de movimiento basándose en una diferencia en una distancia temporal entre el primer vector de movimiento temporal y el segundo vector de movimiento temporal, comprendiendo dicha distancia temporal una distancia de POC entre una imagen de referencia y una imagen objetivo del vector de movimiento temporal respectivo.

30 2. El procedimiento de la reivindicación 1, en el que, cuando el segundo vector de movimiento no es un vector de movimiento temporal, inhabilitar una capacidad de determinar un predictor de vector de movimiento a partir del primer vector de movimiento.

35 3. El procedimiento de la reivindicación 1, que comprende además ubicar el primer bloque identificando una ubicación en una primera imagen que está coubicada con una ubicación del segundo bloque y que se ajusta de acuerdo con una disparidad entre el primer bloque de la primera vista y el segundo de la segunda vista.

40 4. El procedimiento de la reivindicación 1, que comprende además crear una lista de candidatos que contiene datos que identifican una imagen que incluye el segundo bloque de datos de vídeo y uno o más bloques diferentes de datos de vídeo.

45 5. El procedimiento de la reivindicación 4, en el que codificar los datos de predicción comprende decodificar los datos de predicción, en el que decodificar los datos de predicción comprende identificar, a partir de la lista de candidatos, la imagen que incluye el segundo bloque de datos de vídeo, mediante un índice para la lista de candidatos.

6. El procedimiento de la reivindicación 1, en el que el segundo bloque está en la misma ubicación temporal que el primer bloque.

50 7. Un aparato para codificar datos de vídeo, que comprende:

55 medios para identificar un primer bloque de datos de vídeo en una primera ubicación temporal de una primera vista, en el que el primer bloque de datos de vídeo está asociado con un primer vector de movimiento temporal;

60 medios para determinar un predictor de vector de movimiento para un segundo vector de movimiento asociado con un segundo bloque de datos de vídeo, en el que el segundo vector de movimiento es un vector de movimiento temporal y el segundo bloque es de una segunda vista que es diferente de la primera vista, en el que los medios para determinar el predictor de vector de movimiento comprenden medios para usar el primer vector de movimiento como el predictor de vector de movimiento, cuando un valor de recuento de orden de imágenes (POC) de una imagen de referencia del segundo vector de movimiento es igual a un valor de POC de una imagen de referencia del primer vector de movimiento y un valor de POC de una imagen objetivo del segundo vector de movimiento es igual a un valor de POC de una imagen objetivo del primer vector de movimiento; y

65 medios para codificar datos de predicción para el segundo bloque mediante el predictor de vector de

movimiento, estando el aparato caracterizado por que comprende

- 5 medios para escalar el primer vector de movimiento basándose en una diferencia en una distancia temporal entre el primer vector de movimiento temporal y el segundo vector de movimiento temporal, cuando un valor de POC de una imagen de referencia del vector de movimiento para el segundo bloque es diferente de un valor de POC de una imagen de referencia del primer vector de movimiento, comprendiendo dicha distancia temporal una distancia de POC entre una imagen de referencia y una imagen objetivo del vector de movimiento temporal respectivo.
- 10 **8.** El aparato de la reivindicación 6, que comprende además medios para ubicar el primer bloque identificando una ubicación en una primera imagen que está coubicada con una ubicación del bloque para el cual los datos de predicción van a codificarse y que se ajusta de acuerdo con una disparidad entre el primer bloque de la primera vista y el bloque para el cual los datos de predicción de la segunda vista van a codificarse.
- 15 **9.** Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que, al ejecutarse, hacen que uno o más procesadores realicen el procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 6.

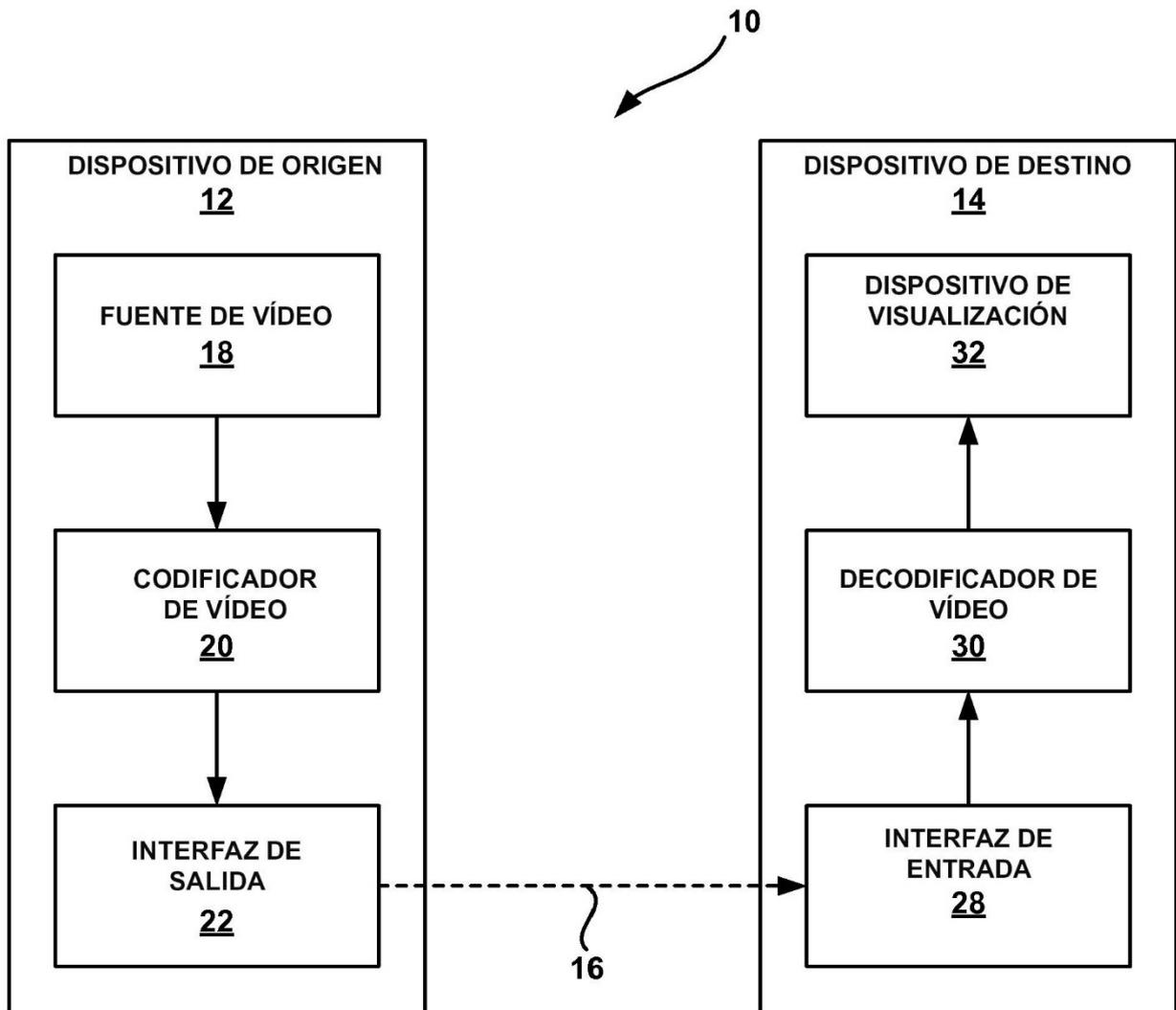


FIG. 1

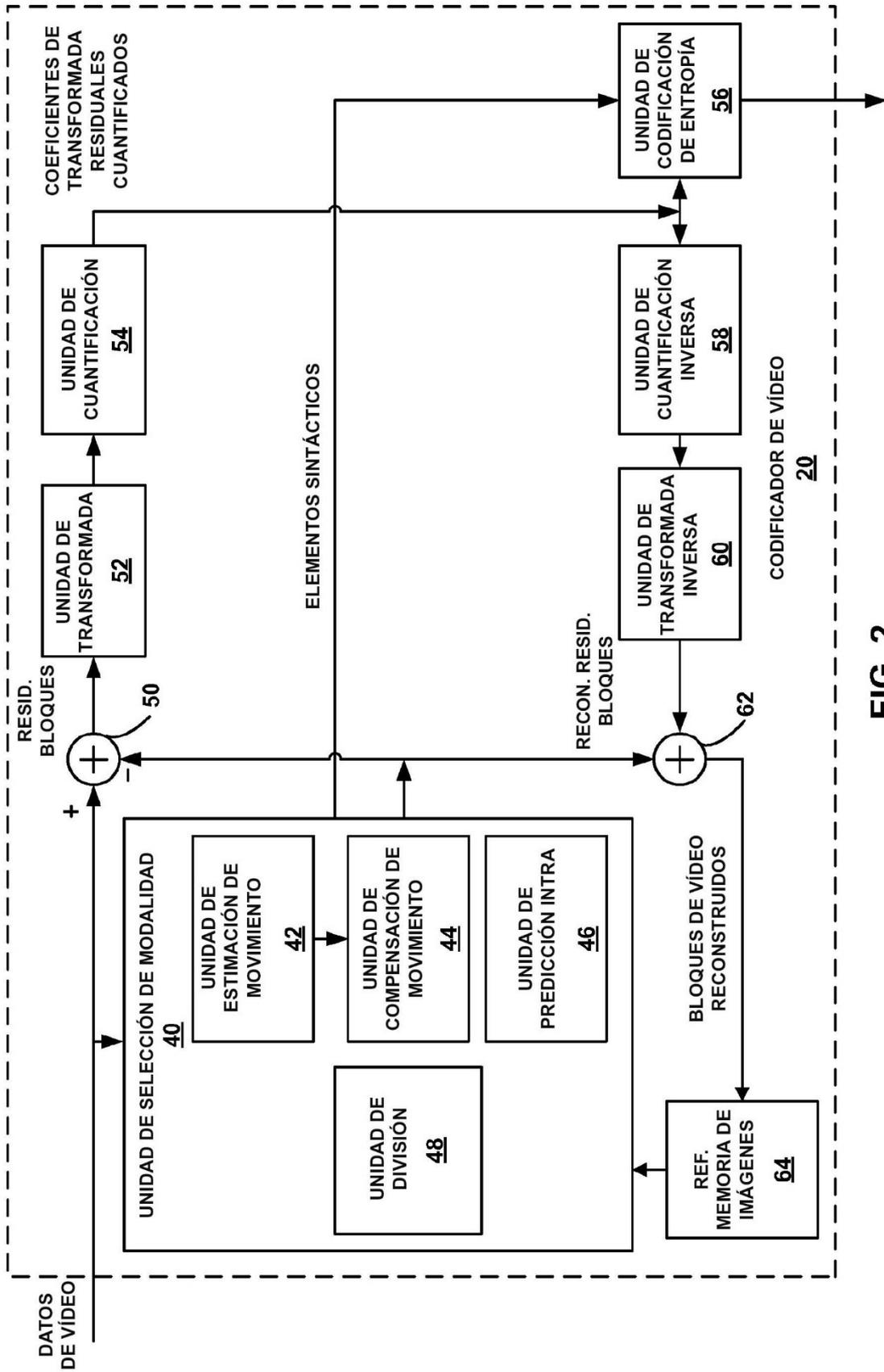


FIG. 2

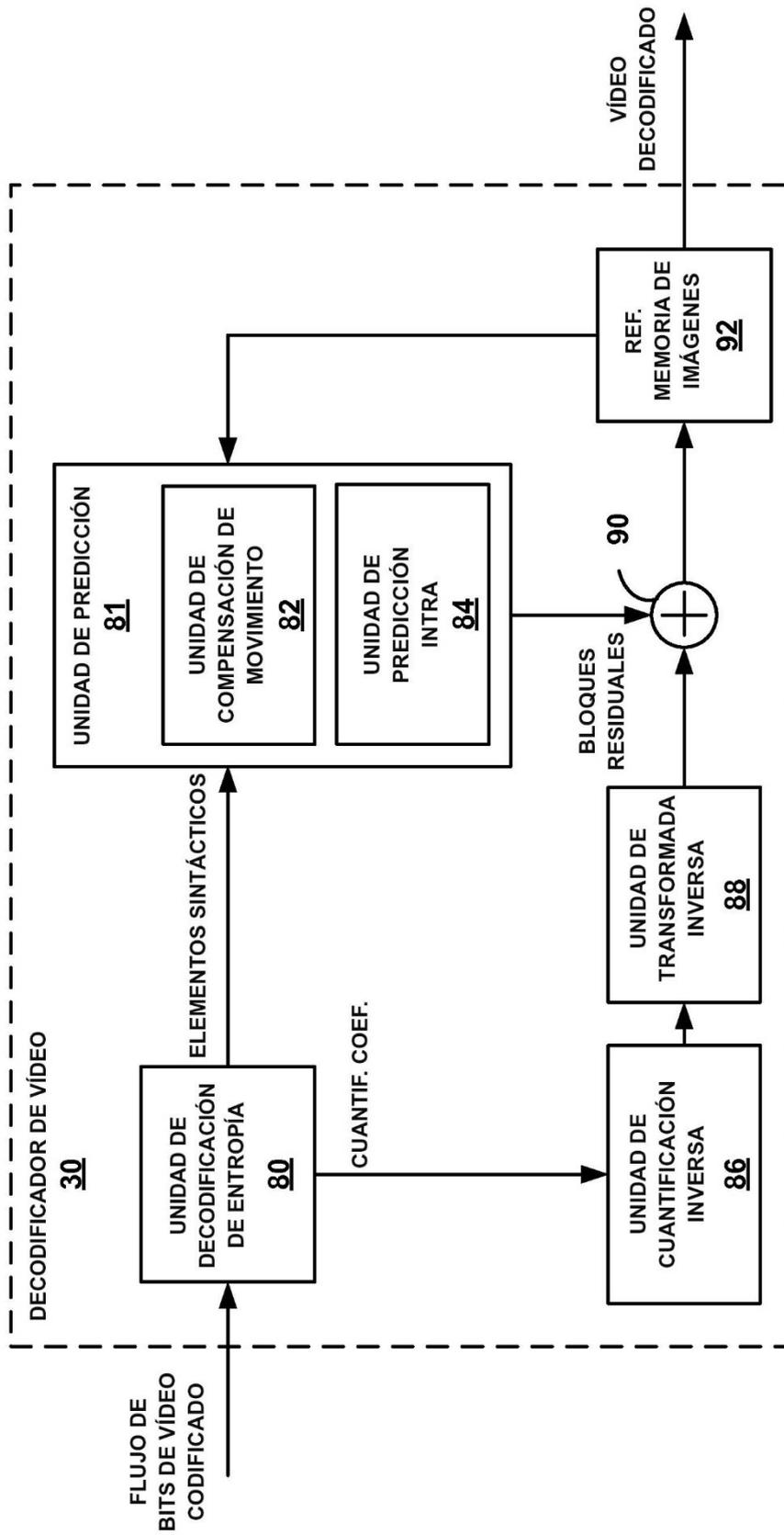


FIG. 3

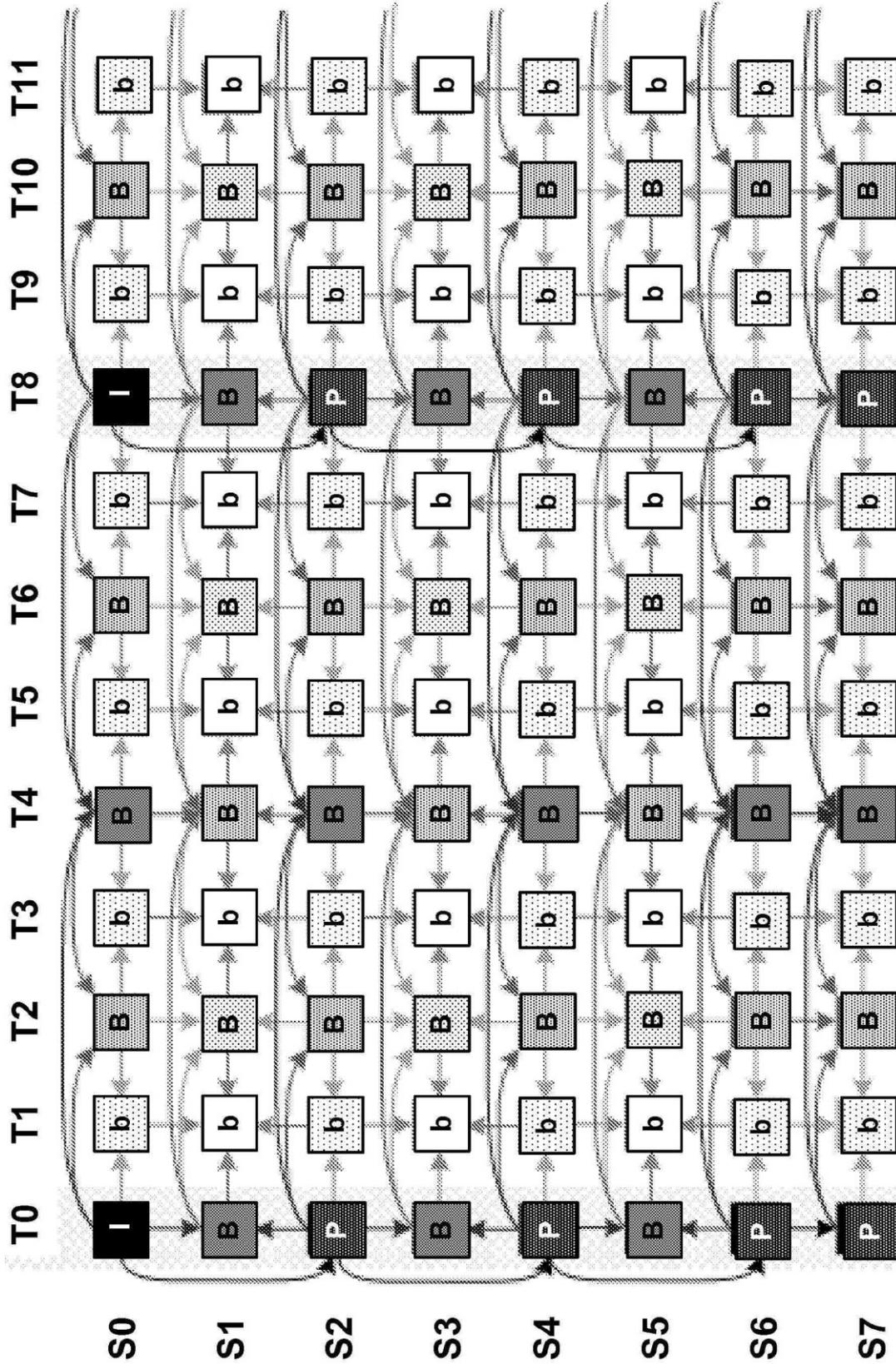


FIG. 4

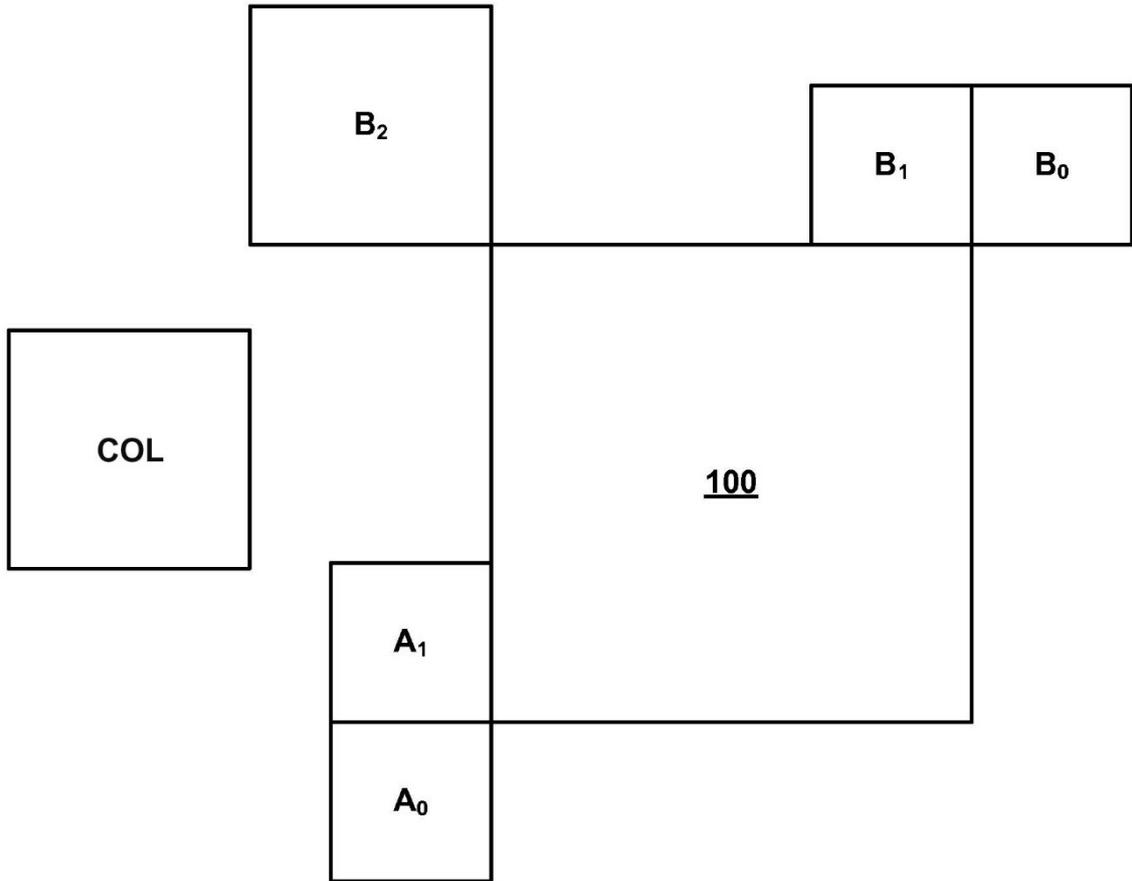


FIG. 5

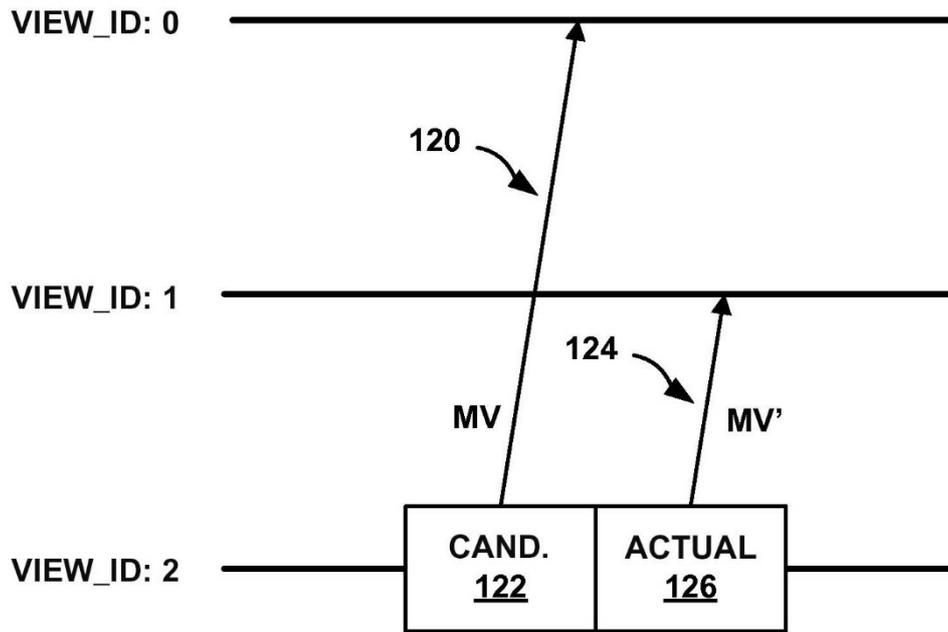


FIG. 6

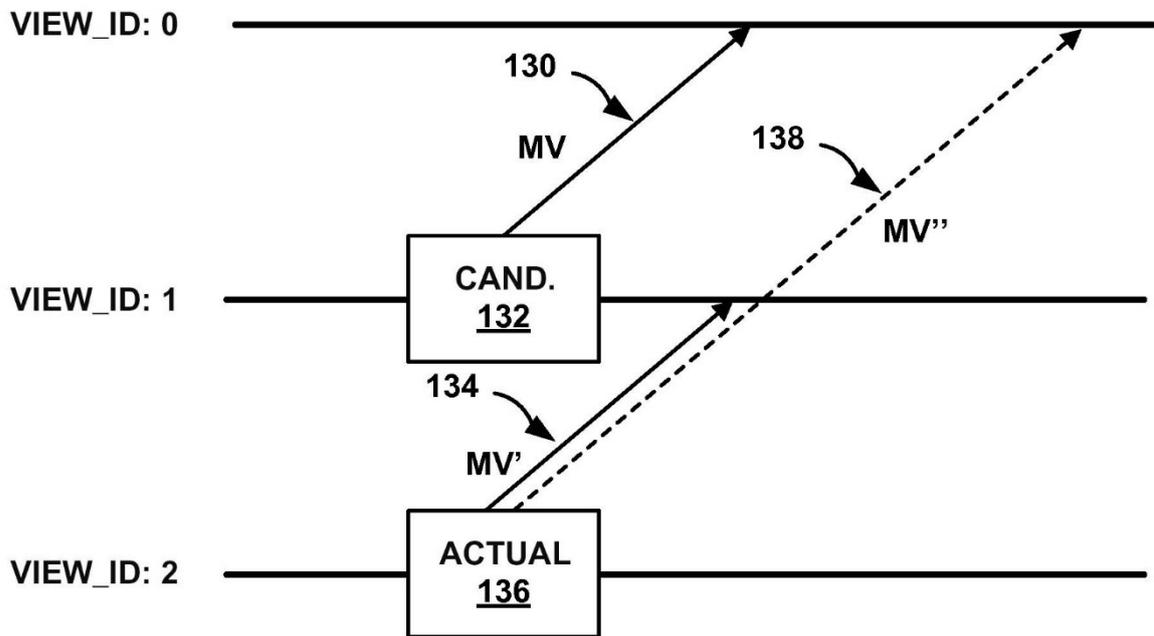


FIG. 7

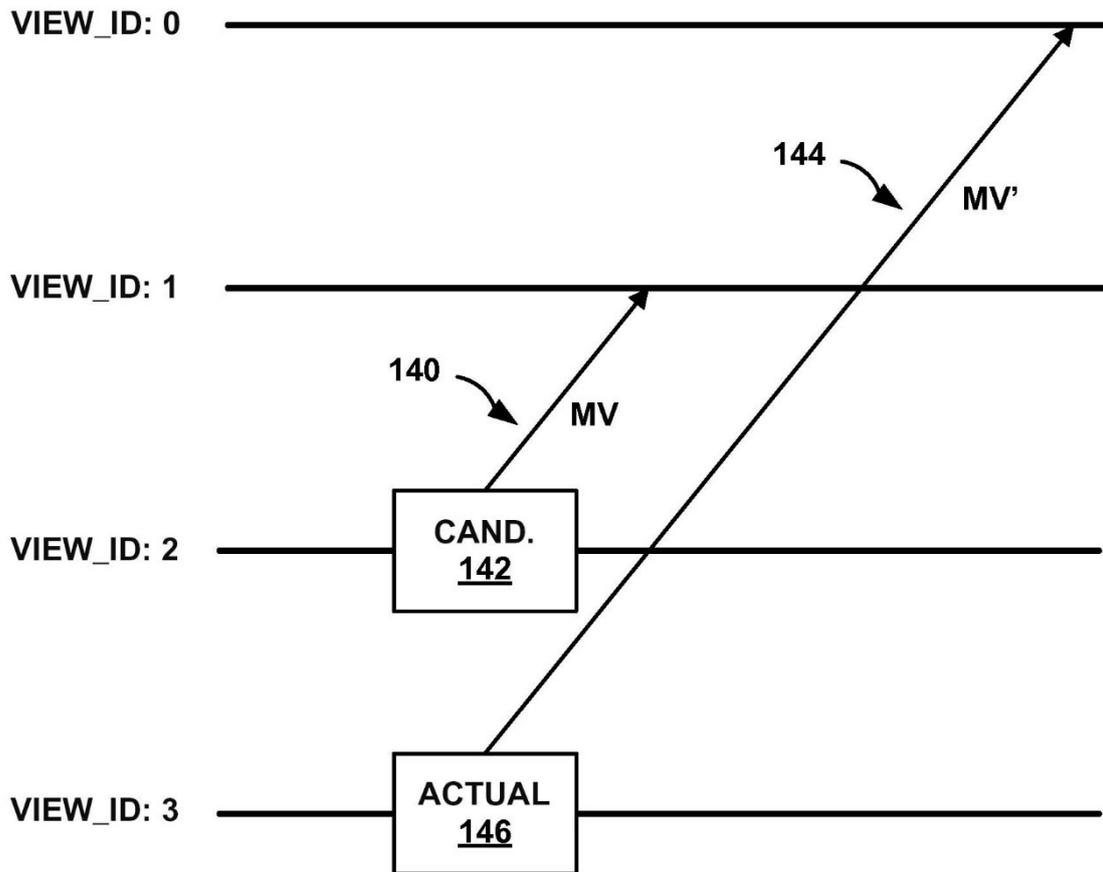


FIG. 8

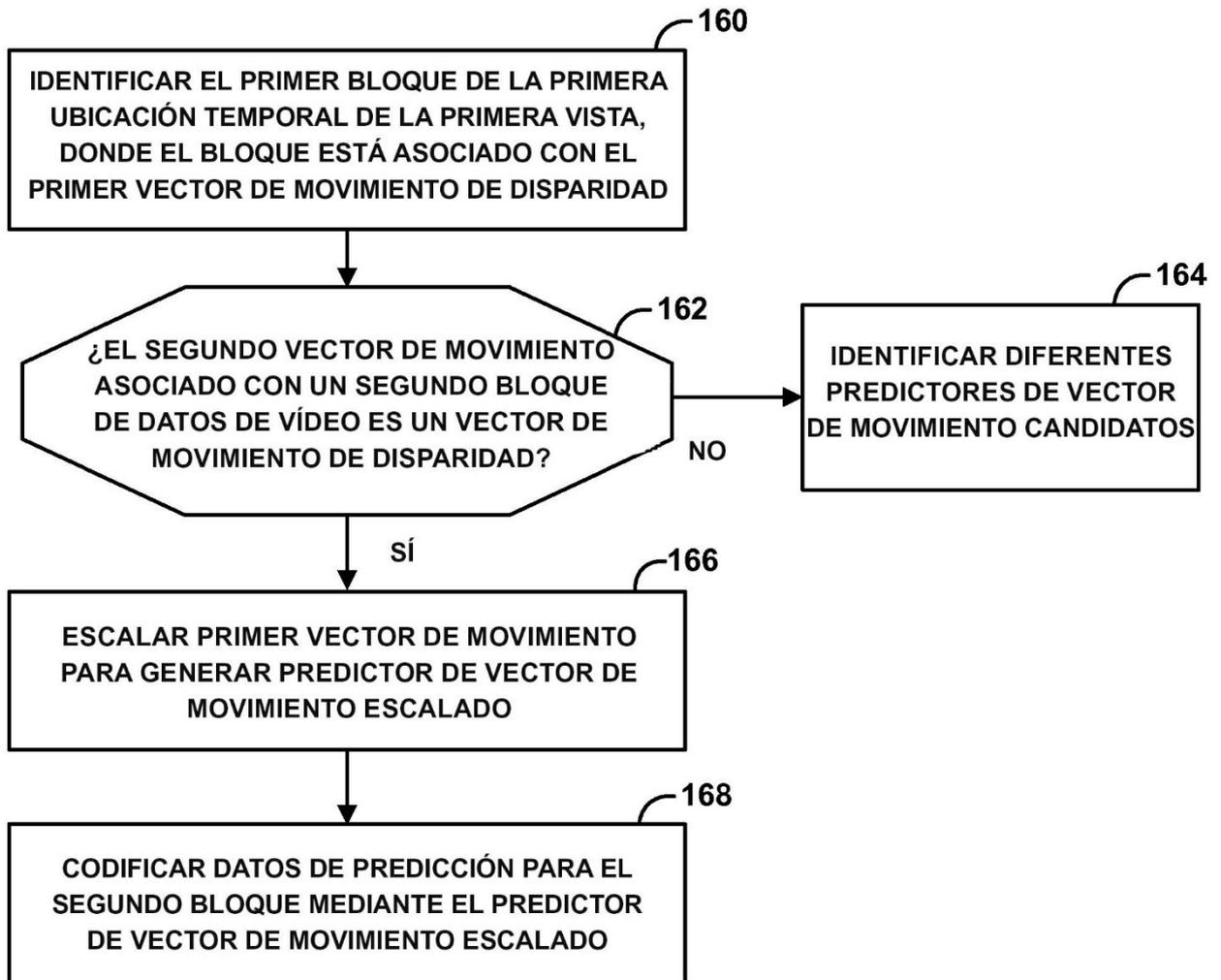


FIG. 9

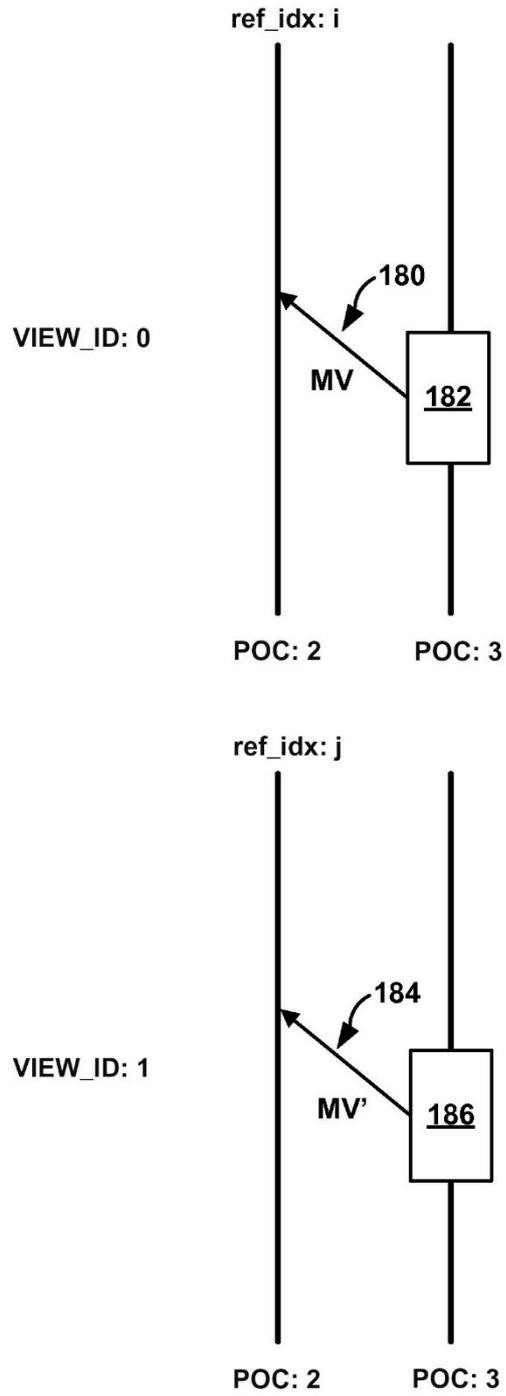


FIG. 10

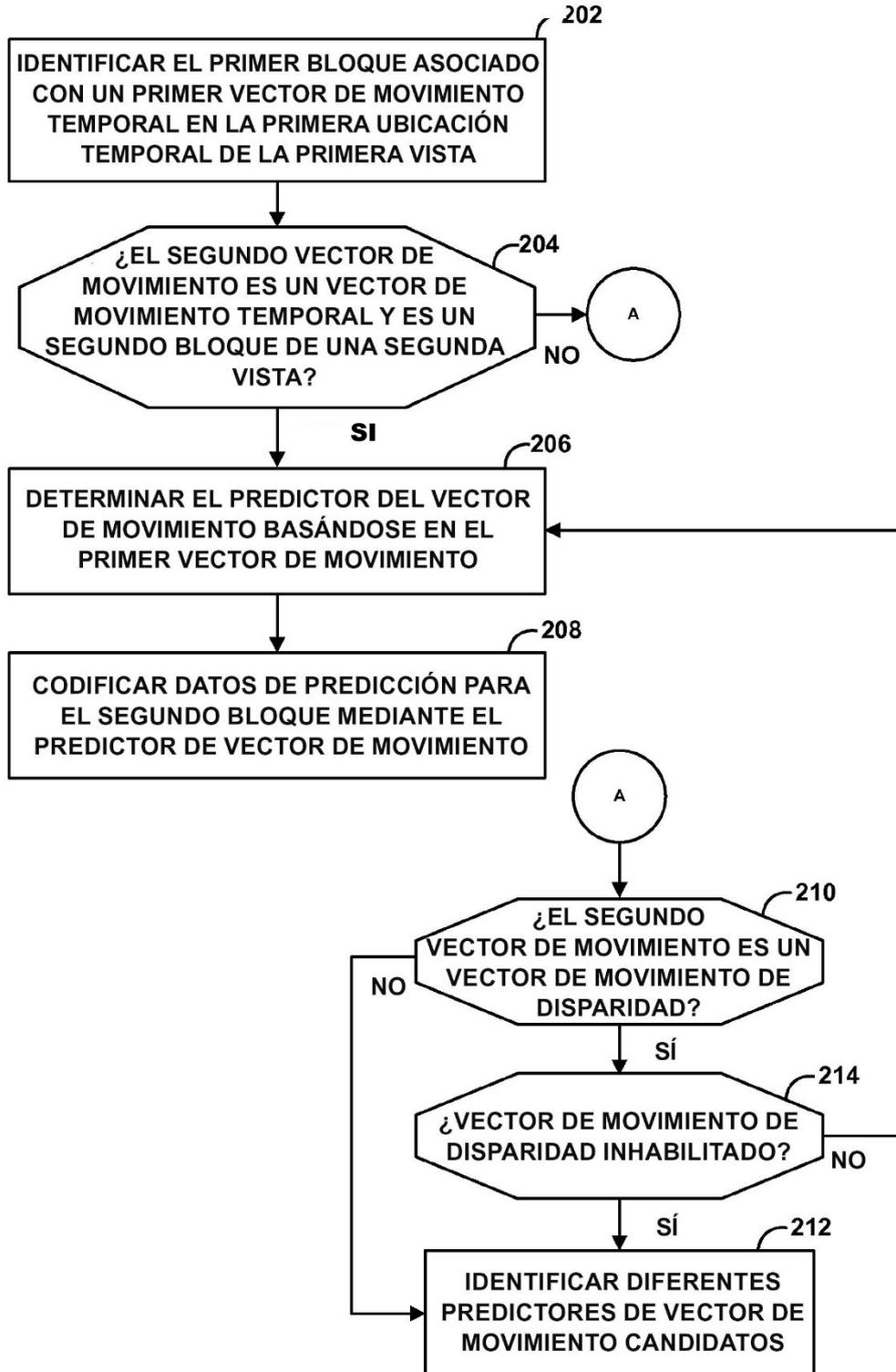


FIG. 11