

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 974 538**

51 Int. Cl.:

H04N 19/187 (2014.01)
H04N 19/597 (2014.01)
H04N 19/70 (2014.01)
H04N 19/30 (2014.01)
H04N 19/105 (2014.01)
H04N 19/107 (2014.01)
H04N 19/61 (2014.01)
H04N 19/136 (2014.01)
H04N 19/103 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **06.12.2013** **PCT/US2013/073638**
87 Fecha y número de publicación internacional: **12.06.2014** **WO14089469**
96 Fecha de presentación y número de la solicitud europea: **06.12.2013** **E 13814698 (0)**
97 Fecha y número de publicación de la concesión europea: **06.03.2024** **EP 2929686**

54 Título: **Predicción residual avanzada en codificación de vídeo escalable y multi-vista**

30 Prioridad:

07.12.2012 US 201261734874 P
05.12.2013 US 201314098333

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
27.06.2024

73 Titular/es:

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

72 Inventor/es:

ZHANG, LI;
CHEN, YING y
KARCZEWICZ, MARTA

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 974 538 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 residual avanzada en codificación de vídeo escalable y multi-vista

5 CAMPO TÉCNICO

La presente divulgación se refiere a la codificación de vídeo.

10 ANTECEDENTES

Las capacidades de vídeo digital se pueden incorporar en una amplia gama de dispositivos, incluidos televisores digitales, sistemas de transmisión directa digital, sistemas de transmisión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas, 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 llamados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo 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 de vídeo avanzada (AVC), la norma de codificación de vídeo de alta eficiencia (HEVC) actualmente en desarrollo y extensiones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital de manera más eficiente mediante la implementación de dichas técnicas de compresión de vídeo.

Las técnicas de compresión de vídeo incluyen predicción espacial (intra-imagen) y/o predicción temporal (inter-imagen) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (es decir, una imagen o una parte de una imagen) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, conjuntos de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican utilizando predicción espacial con respecto a las muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia.

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forma el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una compresión adicional, los datos residuales pueden transformarse del dominio de píxeles a un dominio de transformación, lo que da como resultado coeficientes de transformación residuales, que luego pueden cuantificarse. Los coeficientes de transformación cuantificados, inicialmente dispuestos en una matriz bidimensional, pueden escanearse para producir un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación entrópica para lograr aún más compresión.

Se puede generar un flujo de bits de codificación multi-vista codificando vistas, por ejemplo, desde múltiples perspectivas. Se han desarrollado algunos estándares de vídeo tridimensional (3D) que hacen uso de aspectos de codificación multi-vista. Por ejemplo, diferentes vistas pueden transmitir vistas de los ojos izquierdo y derecho para admitir vídeo en 3D. Como alternativa, algunos procedimientos de codificación de vídeo 3D pueden aplicar la denominada codificación multi-vista más profundidad. En la codificación multi-vista más profundidad, un flujo de bits de vídeo 3D puede contener no solo componentes de vista de textura, sino también componentes de vista de profundidad. Por ejemplo, cada vista puede comprender un componente de vista de textura y un componente de vista de profundidad.

El documento "Test Model under Consideration for HEVC based 3D vídeo coding v.3.0", 100. REUNIÓN MPEG; 30-4-2012 - 4-5-2012; GINEBRA; (GRUPO DE EXPERTOS EN CINE O ISO/IEC JTC/SC29/WG11), no. N12744, 1 de junio de 2012, describe la predicción residual inter-vista, donde un vector de disparidad se determina en función de una estimación de mapa de profundidad asociada con una imagen actual, y un bloque residual en la vista de referencia al que hace referencia el vector de disparidad se utiliza para predecir el residuo del bloque actual.

60 RESUMEN

En general, esta descripción se refiere a la predicción residual inter-vista para codificadores-decodificadores de múltiples capas (códecs) y códecs de vídeo tridimensional (3DV) basados en códecs bidimensionales, tales como la codificación de vídeo de alta eficiencia (HEVC). Las técnicas de esta descripción, en algunos casos, pueden usarse para refinar un procedimiento avanzado de predicción interresidual (ARP). Por ejemplo, los aspectos de esta descripción pueden referirse a habilitar/deshabilitar ARP, interpolación en ARP y factores de ponderación en ARP.

La invención se describe en las reivindicaciones independientes adjuntas. En las reivindicaciones dependientes se definen características opcionales.

- 5 Los detalles de uno o más ejemplos de esta descripción se exponen en los dibujos adjuntos y la descripción a continuación. Otras características, objetos y ventajas serán evidentes a partir de la descripción, los dibujos, y las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

- 10 La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo que puede utilizar las técnicas descritas en esta descripción.
- La Figura 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar las técnicas descritas en esta descripción.
- 15 La Figura 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar las técnicas descritas en esta divulgación.
- La Figura 4 es un diagrama conceptual que ilustra la codificación de una secuencia multi-vista.
- La Figura 5 es un diagrama conceptual que ilustra un patrón de predicción multi-vista de ejemplo.
- La Figura 6 es un diagrama conceptual que ilustra ejemplos de capas escalables de datos de vídeo.
- 20 La Figura 7 es un diagrama conceptual que ilustra conjuntos de predicción (PU) espacialmente vecinos de ejemplo con respecto a un PU actual.
- La Figura 8 es un diagrama de bloques que ilustra la predicción residual inter-vista.
- La Figura 9 es un diagrama conceptual que ilustra una estructura de predicción de ejemplo de predicción residual avanzada (ARP) en codificación de vídeo multi-vista.
- 25 La Figura 10 es un diagrama conceptual que ilustra una relación de ejemplo entre un bloque actual, un bloque de referencia y un bloque compensado por movimiento en ARP.
- La Figura 11 es un diagrama conceptual que ilustra muestras enteras y posiciones de muestras fraccionarias para interpolación luma de un cuarto de muestra.
- La Figura 12 es un diagrama conceptual que ilustra los modos de partición para codificar un bloque de datos de vídeo.
- 30 La Figura 13 es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo, de acuerdo con una o más técnicas de esta descripción.
- La Figura 14 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un decodificador de vídeo, de acuerdo con una o más técnicas de la presente divulgación.
- 35 La Figura 15 es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo, de acuerdo con una o más técnicas de esta descripción.
- La Figura 16 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un decodificador de vídeo, de acuerdo con una o más técnicas de la presente divulgación.
- La Figura 17 es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo, de acuerdo con una o más técnicas de esta descripción.
- 40 La Figura 18 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un decodificador de vídeo, de acuerdo con una o más técnicas de la presente divulgación.
- La Figura 19 es un diagrama de flujo que ilustra una operación de ejemplo de un codificador de vídeo, de acuerdo con una o más técnicas de esta descripción.
- 45 La Figura 20 es un diagrama de flujo que ilustra un ejemplo de funcionamiento de un decodificador de vídeo, de acuerdo con una o más técnicas de la presente divulgación.

DESCRIPCIÓN DETALLADA

- 50 Las técnicas de esta divulgación generalmente se refieren a diversas técnicas para mejorar aún más la eficiencia de codificación de predicción residual avanzada (ARP) para códecs multi-vista, 3DV (por ejemplo, multi-vista más profundidad) o escalables basados en códecs bidimensionales (2D) avanzados. Por ejemplo, el Equipo Conjunto de Colaboración sobre Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo del UIT-T (VCEG) y el Grupo de Expertos en Películas de ISO/IEC (MPEG) están desarrollando un estándar de codificación de vídeo de alta eficiencia (HEVC). Un borrador del estándar HEVC, denominado "Borrador de trabajo HEVC 9" (también denominado en esta invención WD9) se describe en Bross y col., "High Efficiency Video Coding (HEVC) text specification draft 9," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 11th Meeting, Shanghai, China, octubre de 2012 y está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v10.zip.
- 60 De manera similar a los proyectos basados en AVC, un Equipo Conjunto de Colaboración en Codificación de Vídeo 3D (JCT-3V) de VCEG y MPEG está llevando a cabo el estudio de dos soluciones 3DV que utilizan la tecnología de codificación HEVC. La primera es la extensión multi-vista de HEVC, denominada MV-HEVC y otra es el códec 3DV completo basado en HEVC de profundidad mejorada, 3D-HEVC. Parte de los esfuerzos de estandarización incluye la estandarización de la codificación de vídeo multi-vista/3D basada en HEVC. La última versión del software 3D-HTM
- 65

5.0 está disponible electrónicamente en https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-5.0/. Las técnicas descritas a continuación pueden implementarse junto con las dos soluciones 3DV propuestas anteriormente.

En algunos casos, las técnicas también pueden (o alternativamente) implementarse con una extensión escalable a HEVC. En la codificación de vídeo escalable, se pueden codificar múltiples capas de datos de vídeo. En algunos casos, cada capa puede corresponder a una vista particular. Aquí, la aplicación de la escalabilidad de la vista y la escalabilidad espacial se consideran altamente beneficiosas en la evolución de los servicios 3D, ya que permiten extensiones compatibles con versiones anteriores para más vistas y/o mejoran la resolución de las vistas de manera que es posible la decodificación por dispositivos heredados.

En la codificación de vídeo bidimensional, los datos de vídeo (es decir, una secuencia de imágenes) se codifican imagen por imagen, no necesariamente en orden de visualización. Los dispositivos de codificación de vídeo dividen cada imagen en bloques y codifican cada bloque individualmente. Los modos de predicción basados en bloques incluyen la predicción espacial, también denominada intrapredicción, y la predicción temporal, también denominada interpredicción.

Para datos de vídeo tridimensionales, tales como datos codificados multi-vista o escalables, los bloques también se pueden predecir inter-vista y/o inter-capa. Como se describe en esta invención, una "capa" de vídeo puede referirse generalmente a una secuencia de imágenes que tienen al menos una característica común, tal como una vista, una velocidad de fotogramas, una resolución o similares. Por ejemplo, una capa puede incluir datos de vídeo asociados con una vista particular (por ejemplo, perspectiva) de datos de vídeo de múltiples vistas. Como otro ejemplo, una capa puede incluir datos de vídeo asociados con una capa particular de datos de vídeo escalables.

Por lo tanto, esta divulgación puede referirse indistintamente a una capa y una vista de datos de vídeo. Es decir, una vista de datos de vídeo puede denominarse como una capa de datos de vídeo y viceversa. Además, los términos predicción inter-vista y predicción inter-capa pueden referirse indistintamente a la predicción entre múltiples capas y/o vistas de datos de vídeo. Además, el códec multicapa (o codificador de vídeo multicapa) puede referirse conjuntamente a un códec multi-vista o un códec escalable.

En la codificación de vídeo multi-vista o escalable, los bloques se pueden predecir a partir de una imagen de otra vista o capa de datos de vídeo. De esta manera, se puede habilitar la predicción inter-vista basada en componentes de vista reconstruidos a partir de diferentes vistas. Esta divulgación utiliza el término "componente de vista" para referirse a una imagen codificada de una vista o capa particular. Es decir, un componente de vista puede comprender una imagen codificada para una vista particular en un momento particular (en términos de orden de visualización u orden de salida). Un componente de vista (o fragmentos de un componente de vista) puede tener un valor de recuento de orden de imagen (POC), que generalmente indica el orden de visualización (u orden de salida) del componente de vista.

Por lo general, los mismos o correspondientes objetos de dos vistas no están ubicados conjuntamente. 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 con respecto al objeto correspondiente en una vista diferente. Dicho vector también puede denominarse como un "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 en una imagen de una primera vista puede desplazarse con respecto a un píxel correspondiente en una imagen de una segunda vista mediante un vector de disparidad particular relacionado con diferentes ubicaciones de cámara desde las que se capturan la primera vista y la segunda vista. En algunos ejemplos, el vector de disparidad se puede usar para predecir la información de movimiento (vector(es) de movimiento con o sin índice(s) de imagen de referencia) de una vista a otra vista.

Por lo tanto, para mejorar aún más la eficiencia de codificación, un codificador de vídeo también puede aplicar predicción de movimiento inter-vista y/o predicción residual inter-vista. Con respecto a la predicción de movimiento inter-vista, un codificador de vídeo puede codificar un vector de movimiento asociado con un bloque de una vista con respecto a un vector de movimiento asociado con un bloque de una segunda vista diferente. Del mismo modo, como se describe con mayor detalle a continuación, en la predicción residual inter-vista, un codificador de vídeo puede codificar datos residuales de una vista en relación con el residuo de una segunda vista diferente. En algunos casos, la predicción residual inter-vista puede denominarse predicción residual avanzada (ARP), particularmente en el contexto de 3D-HEVC.

En ARP, un codificador de vídeo determina un bloque predictivo para predecir un bloque actual. El bloque predictivo para el bloque actual puede basarse en muestras de una imagen de referencia temporal que están asociadas con una ubicación indicada por un vector de movimiento del bloque actual. La imagen de referencia temporal está asociada con la misma vista que la imagen actual, pero está asociada con una instancia de tiempo diferente a la imagen actual. En algunos casos, cuando las muestras de un bloque se basan en muestras de una imagen particular, las muestras pueden basarse en muestras reales o interpoladas de la imagen particular.

Además, en ARP, el codificador de vídeo determina un bloque de referencia de disparidad en función de muestras de

una imagen de referencia de disparidad que se encuentran en una ubicación indicada por un vector de disparidad del bloque actual. La imagen de referencia de disparidad está asociada con una vista diferente (es decir, una vista de referencia) que la imagen actual, pero está asociada con la misma instancia de tiempo que la imagen actual.

El codificador de vídeo también determina un bloque de referencia de disparidad temporal para el bloque actual. El bloque de referencia temporal se basa en muestras de una imagen de referencia de disparidad temporal que están asociadas con una ubicación indicada por el vector de movimiento del bloque actual y el vector de disparidad. Por ejemplo, el bloque de referencia de disparidad temporal puede ubicarse aplicando el vector de movimiento temporal (por ejemplo, reutilizando el vector de movimiento temporal) al bloque de referencia de disparidad. Por lo tanto, la imagen de referencia de disparidad temporal está asociada con la misma vista que la imagen de referencia de disparidad y está asociada con el mismo conjunto de acceso que la imagen de referencia temporal del bloque actual.

Si bien el bloque de referencia de disparidad temporal se describe en esta invención como ubicado mediante la aplicación del vector de movimiento temporal al bloque de referencia de disparidad con fines ilustrativos, en algunos casos, el vector de movimiento temporal puede no aplicarse directamente a la imagen de referencia de disparidad. Más bien, el vector de movimiento temporal puede combinarse con el vector de disparidad para localizar el bloque de referencia de disparidad temporal, por ejemplo, en relación con el bloque actual. Por ejemplo, supongamos con fines ilustrativos que los vectores de disparidad se denotan como DV[0] y DV[1] y los vectores de movimiento temporal se denotan como TMV[0] y TMV[1]. En este ejemplo, un codificador de vídeo (como un codificador de vídeo o un decodificador de vídeo) puede determinar la ubicación de los bloques de disparidad temporal en las imágenes de referencia de disparidad temporal en relación con el bloque actual combinando los vectores de disparidad y los vectores de movimiento temporal, por ejemplo, DV[0]+TMV[0], DV[1]+TMV[1]. Por lo tanto, las referencias en esta invención a "aplicar un vector de movimiento temporal a un bloque de referencia de disparidad" no necesariamente requieren que el vector de movimiento temporal se aplique directamente a la ubicación del bloque de referencia de disparidad.

El codificador de vídeo determina entonces un predictor residual para predecir el residuo asociado con el bloque actual, por ejemplo, la diferencia entre el bloque actual y el bloque de referencia temporal. Cada muestra del predictor residual para el bloque actual indica una diferencia entre una muestra del bloque de referencia de disparidad y una muestra correspondiente del bloque de referencia de disparidad temporal. En algunos casos, el codificador de vídeo puede aplicar un factor de ponderación (por ejemplo, 0, 0,5, 1 o similar) al predictor residual para aumentar la precisión del predictor residual.

En los casos en los que el codificador de vídeo es un codificador de vídeo, el codificador de vídeo puede determinar un bloque residual final para el bloque actual. El bloque residual final comprende muestras que indican diferencias entre muestras del bloque actual, muestras en el bloque predictivo temporal y muestras en el predictor residual. El codificador de vídeo puede incluir, en un flujo de bits, datos que representan el bloque residual final. En los casos en los que el codificador de vídeo es un decodificador de vídeo, el decodificador de vídeo puede reconstruir el bloque actual en función del bloque residual final, el predictor residual y el bloque predictivo temporal.

Si bien ARP puede mejorar la eficiencia de codificación de la predicción residual inter-vista (o inter-capas), son posibles más refinamientos. Por ejemplo, determinadas técnicas de esta divulgación se refieren al factor de ponderación ARP. Como se señaló anteriormente, un codificador de vídeo puede aplicar un factor de ponderación al predictor residual. En general, el factor de ponderación siempre se señala en el flujo de bits, independientemente de si hay una imagen de referencia temporal en las listas de imágenes de referencia para codificar el bloque actual. Sin embargo, la señalización del factor de ponderación cuando no hay una imagen de referencia temporal puede aumentar innecesariamente la complejidad y disminuir la eficiencia, porque si no hay imágenes de referencia temporales, no hay predicción temporal y residuo asociado para aplicar ARP.

Un caso en el que puede no haber imágenes de referencia temporales en las listas de imágenes de referencia (por ejemplo, ni lista 0 ni lista 1) es cuando se codifican imágenes de acceso aleatorio. Como se describe con mayor detalle a continuación, las imágenes de acceso aleatorio no se predicen temporalmente. Las imágenes de acceso aleatorio generalmente solo se predicen intra o se predicen inter-vista (solo las imágenes de referencia inter-vista se incluyen en una lista de imágenes de referencia). Por lo tanto, como se señaló anteriormente, la señalización de los factores de ponderación es innecesaria e ineficiente, porque no hay un residuo para el cual determinar un predictor.

De acuerdo con los aspectos de esta divulgación, un codificador de vídeo (tal como un codificador de vídeo o un decodificador de vídeo) puede habilitar o deshabilitar ARP (incluida la codificación residual de una capa en relación con el residuo de una segunda capa diferente) en función de las imágenes de referencia en las listas de imágenes de referencia para el bloque que se está codificando actualmente. En un ejemplo, el codificador de vídeo puede habilitar o deshabilitar ARP en función de si las listas de imágenes de referencia (por ejemplo, lista 0 o lista 1) para el bloque que se está codificando actualmente incluyen imágenes de referencia temporales. De acuerdo con los aspectos de esta divulgación, si las listas de imágenes de referencia para un fragmento inter-predicho incluyen solo imágenes de referencia inter-vista, el codificador de vídeo puede deshabilitar ARP al codificar los bloques del fragmento. En dicho ejemplo, cuando el codificador de vídeo comprende un codificador de vídeo, el codificador de vídeo puede no señalar

un factor de ponderación para todos los bloques (por ejemplo, conjuntos de codificación o conjuntos de predicción en el contexto de la codificación de vídeo de alta eficiencia (HEVC), como se describe con mayor detalle a continuación) dentro del fragmento en el flujo de bits (omitir la señalización de un factor de ponderación). Del mismo modo, cuando el codificador de vídeo comprende un decodificador de vídeo, el decodificador de vídeo también puede omitir la decodificación de un factor de ponderación y determinar automáticamente (es decir, inferir) que el factor de ponderación es igual a cero.

Las técnicas descritas anteriormente se pueden aplicar en el contexto de imágenes de acceso aleatorio. Por ejemplo, de acuerdo con los aspectos de esta divulgación, un codificador de vídeo puede habilitar o deshabilitar ARP en función de si el componente de vista que se está codificando actualmente es un componente de vista de acceso aleatorio. Como se señaló anteriormente, los componentes de vista de acceso aleatorio no tienen imágenes de referencia temporales, ya que las imágenes de acceso aleatorio se intra-predicen o se predicen inter-vista. Por lo tanto, el codificador de vídeo puede desactivar ARP para cada bloque de un componente de vista de acceso aleatorio. Una vez más, el codificador de vídeo puede no señalar un factor de ponderación en el flujo de bits (omitir la señalización de un factor de ponderación). Del mismo modo, cuando el codificador de vídeo comprende un decodificador de vídeo, el decodificador de vídeo también puede omitir la decodificación de un factor de ponderación e inferir que el factor de ponderación es igual a cero.

En otro ejemplo, de acuerdo con aspectos de esta descripción, un codificador de vídeo puede habilitar ARP si al menos una imagen de referencia es de la misma vista que el bloque que se está codificando actualmente. Adicional o alternativamente, el codificador de vídeo puede habilitar ARP solo cuando ambas imágenes de referencia (correspondientes a una imagen de referencia en RefPicList0 y una imagen de referencia en RefPicList1), si están disponibles, son de la misma vista que el bloque que se está codificando actualmente. Adicional o alternativamente, el codificador de vídeo puede deshabilitar ARP para un bloque si el bloque está codificado inter-vista con una imagen de referencia inter-vista. Como se señaló anteriormente, cuando ARP está desactivado, el factor de ponderación no se señala.

Las técnicas de esta descripción también se refieren a la interpolación en ARP. Por ejemplo, cuando se realiza ARP (por ejemplo, el factor de ponderación no es cero) tanto un codificador de vídeo como un decodificador de vídeo pueden usar un procedimiento de compensación de movimiento adicional durante el procedimiento de generación de predictores residuales. Por lo tanto, si un vector de movimiento indica una ubicación de píxel fraccional (pel fraccional), el codificador de vídeo realiza dos procedimientos de interpolación de pel fraccional, por ejemplo, un procedimiento de interpolación para ubicar el bloque de referencia temporal y otro procedimiento de interpolación para ubicar el bloque de referencia de disparidad-temporal. Además, el codificador de vídeo puede aplicar otro procedimiento de interpolación de pel fraccional al determinar un bloque de referencia de disparidad. En HEVC, se especifica un filtro de 8 tomas para los componentes de luminancia, mientras que se especifica un filtro de 4 tomas para los componentes de crominancia. Dichos procedimientos de interpolación pueden aumentar la complejidad informática asociada con ARP.

De acuerdo con los aspectos de esta descripción, el procedimiento de compensación de movimiento de ARP puede simplificarse, particularmente con respecto a la interpolación de subpíxeles (sub-pel) de bloques de referencia. Por ejemplo, un codificador de vídeo puede determinar el bloque de referencia de disparidad de una manera similar o igual al procedimiento utilizado para generar la señal de predicción durante la compensación de movimiento (por ejemplo, el procedimiento utilizado para determinar el bloque de referencia temporal). Es decir, el codificador de vídeo puede determinar el bloque de referencia de disparidad usando la imagen de vista de referencia reconstruida con un vector de disparidad del bloque actual.

En algunos casos, de acuerdo con aspectos de la presente divulgación, el codificador de vídeo puede usar uno o más tipos de interpolaciones para determinar las ubicaciones de los bloques de referencia en ARP. Por ejemplo, el codificador de vídeo puede usar un filtro de paso bajo, tal como un filtro bilineal, para interpolar la ubicación del bloque de referencia de disparidad. Adicional o alternativamente, el codificador de vídeo puede usar el filtro de paso bajo para interpolar la ubicación del bloque de referencia de disparidad temporal. En otro ejemplo más, el codificador de vídeo puede usar el filtro de paso bajo para interpolar la ubicación del bloque de referencia temporal. Por consiguiente, de acuerdo con aspectos de esta descripción, el codificador de vídeo puede usar un filtro bilineal para interpolar la ubicación de uno o más bloques de referencia en ARP, que puede ser más eficiente computacionalmente que aplicar los filtros de derivación más altos especificados por HEVC. Si bien en esta invención se hacen referencias a filtros bilineales, debe entenderse que también o alternativamente pueden usarse uno o más filtros de paso bajo. De acuerdo con los aspectos de la presente divulgación, el codificador de vídeo puede aplicar los filtros de paso bajo descritos anteriormente a cualquier combinación de componentes de luminancia, componentes de crominancia o componentes de luminancia y crominancia.

Las técnicas de esta divulgación también se refieren a la señalización de un factor de ponderación ARP para modos de codificación y/o modos de partición particulares. Por ejemplo, en general, se puede señalar un factor de ponderación para todos los modos de partición (como se describe con mayor detalle, por ejemplo, con respecto al ejemplo que se muestra en la Figura 12) que incluyen PART_2Nx2N, PART_2NxN, PART_Nx2N o similares, y todos

los modos intercodificados que incluyen salto, fusión, predicción avanzada de vectores de movimiento (AMVP). Señalar el factor de ponderación para todos los modos de partición e intermodos puede aumentar innecesariamente la complejidad y disminuir la eficiencia, porque ARP puede no aplicarse de manera eficiente con ciertos modos de partición o intermodos.

De acuerdo con aspectos de esta descripción, ARP puede habilitarse o deshabilitarse en función de un modo de partición y/o modo de codificación del bloque que se está codificando actualmente. Por ejemplo, los factores de ponderación solo pueden señalizarse para ciertos modos de partición y/o ciertos modos de codificación. Si un factor de ponderación no se incluye en un flujo de bits, un decodificador de vídeo puede omitir la decodificación del factor de ponderación e inferir que el factor de ponderación es de valor cero (desactivando así ARP). De acuerdo con aspectos de esta descripción, en algunos ejemplos, el factor de ponderación para cualquier bloque intercodificado con modo de partición desigual a PART_2Nx2N puede no señalizarse. En otro ejemplo, puede no señalizarse el factor de ponderación para un bloque intercodificado con un modo de partición distinto de PART_2Nx2N, PART_2NxN y PART_Nx2N. En incluso otro ejemplo, adicional o alternativamente, el factor de ponderación para cualquier bloque intercodificado con modo de codificación desigual al modo de salto y/o modo de fusión puede no señalizarse.

Las técnicas de la presente descripción también se refieren a refinar la manera en que se señalan los factores de ponderación en el flujo de bits. Por ejemplo, en general, un codificador de vídeo puede seleccionar un factor de ponderación de un conjunto fijo de tres factores de ponderación fijos (por ejemplo, 0, 0,5 y 1). Sin embargo, en algunos casos, tres factores de ponderación estáticos pueden no proporcionar suficiente flexibilidad para lograr una eficiencia de predicción suficiente, debido a las diferencias de calidad entre una vista actual y su vista de referencia. Las diferencias de calidad entre la vista actual y la vista de referencia pueden ser dinámicas, en particular con respecto a la codificación de vídeo escalable. Por el contrario, tres factores de ponderación pueden exceder las necesidades de algunos fragmentos o imágenes. Es decir, es posible que algunos fragmentos o imágenes no tengan que seleccionar entre tres factores de ponderación para lograr un equilibrio óptimo entre la complejidad y la mejora de la eficiencia de la codificación.

De acuerdo con los aspectos de esta descripción, se puede implementar un enfoque más flexible para los factores de ponderación. Por ejemplo, la cantidad de factores de ponderación disponibles puede alterarse a nivel de secuencia (por ejemplo, en un conjunto de parámetros, tal como un conjunto de parámetros de secuencia (SPS)). En un ejemplo con fines ilustrativos, un indicador puede señalizarse en un SPS para desactivar uno o más factores de ponderación, por ejemplo, 0,5 y/o 1. En otro ejemplo, dicho indicador puede señalizarse en un conjunto de parámetros de vídeo (VPS) y aplicarse a todas las vistas sin base. En otro ejemplo más, dicho indicador puede señalizarse en una extensión de VPS para cada vista sin base. En otro ejemplo, dicho indicador puede proporcionarse en un conjunto de parámetros de imagen (PPS), un encabezado de fragmento o un conjunto de parámetros de vista para deshabilitar uno o más factores de ponderación. Cuando se ha deshabilitado un factor de ponderación, se pueden usar menos bits para representar los factores de ponderación restantes, proporcionando así un ahorro de bits.

Según otros aspectos, se puede proporcionar un indicador para modificar y/o reemplazar uno o más factores de ponderación. En un ejemplo, el codificador de vídeo puede reemplazar el factor de ponderación de 0,5 por un factor de ponderación de 0,75. Este indicador puede señalizarse en un encabezado de fragmento, un SPS, un conjunto de parámetros de imagen (PPS) o un VPS.

Las técnicas de esta descripción también se refieren a determinar si habilitar o deshabilitar un procedimiento ARP en función de las imágenes de referencia de un búfer de imágenes decodificadas (que también puede denominarse indistintamente memoria de imágenes de referencia, como se describe con mayor detalle con respecto a las Figuras 2 y 3 a continuación) y/o listas de imágenes de referencia. Por ejemplo, como se señaló anteriormente, un bloque de referencia de disparidad temporal para determinar el predictor residual se ubica típicamente aplicando el vector de movimiento temporal al bloque de referencia de disparidad. Sin embargo, en algunos casos, el búfer de imágenes decodificadas puede no contener la imagen indicada mediante la aplicación del vector de movimiento temporal al bloque de referencia de disparidad. Es decir, el búfer de imágenes decodificadas puede no contener una imagen en la misma vista que el bloque de referencia de disparidad que también tenga el mismo valor de recuento de orden de imágenes (POC) que la imagen de referencia temporal del bloque actual.

En algunos ejemplos, incluso si la imagen se incluye en el búfer de imágenes decodificadas, la lista de imágenes de referencia o las listas de imágenes de referencia del fragmento que contiene el bloque de referencia de disparidad pueden no contener la imagen indicada aplicando el vector de movimiento temporal al bloque de referencia de disparidad, por ejemplo, la imagen de referencia de disparidad temporal potencial. En tales casos, la ubicación del bloque de referencia de disparidad temporal puede introducir un error y/o retraso en el procedimiento de codificación.

De acuerdo con los aspectos de esta divulgación, un codificador de vídeo puede habilitar o deshabilitar ARP en función de las imágenes de un búfer de imágenes decodificadas y/o listas de imágenes de referencia. Por ejemplo, cuando el búfer de imágenes decodificadas para codificar un bloque actual no incluye una imagen en la misma vista que la imagen de referencia de disparidad que tiene el mismo POC que la imagen de referencia temporal del bloque actual, un codificador de vídeo puede modificar el procedimiento ARP.

En otro ejemplo, adicional o alternativamente, cuando la o las listas de imágenes de referencia del bloque de referencia de disparidad no incluyen una imagen en la misma vista que la imagen de referencia de disparidad que tiene el mismo POC que la imagen de referencia temporal del bloque actual, un codificador de vídeo puede modificar el procedimiento ARP. Es decir, dado el índice de lista de imágenes de referencia actual X (siendo X 0 o 1), en un ejemplo, si la lista de imágenes de referencia con un índice de lista igual a X del bloque de referencia de disparidad no incluye una imagen de referencia en la misma vista que la imagen de referencia de disparidad y que tiene el mismo POC que la imagen de referencia temporal del bloque actual, el codificador de vídeo puede modificar el procedimiento ARP. En otro ejemplo, si ninguna de las listas de imágenes de referencia (por ejemplo, ni la lista 0 ni la lista 1) del bloque de referencia de disparidad incluye una imagen de referencia en la misma vista que la imagen de referencia de disparidad y que tiene el mismo POC que la imagen de referencia temporal del bloque actual, el codificador de vídeo puede modificar el procedimiento ARP.

En algunos ejemplos, el codificador de vídeo puede modificar el procedimiento ARP deshabilitando el procedimiento ARP, de modo que el bloque actual no esté codificado usando ARP. Es decir, el predictor residual no se genera o siempre se establece en 0. En otros ejemplos, el codificador de vídeo puede modificar el procedimiento ARP escalando el vector de movimiento temporal para identificar otra imagen de referencia de disparidad temporal. Por ejemplo, el codificador de vídeo puede escalar el vector de movimiento temporal de modo que, cuando se aplica a la imagen de referencia de disparidad, el vector de movimiento escalado identifica una imagen de referencia de disparidad temporal que se incluye en la lista de imágenes de referencia y está en una ubicación temporalmente más cercana a la imagen de referencia de disparidad. Las técnicas descritas anteriormente pueden evitar que el codificador de vídeo intente localizar el bloque de referencia de disparidad en una imagen que no está incluida en la o las listas de imágenes de referencia.

La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo 10 que puede utilizar las técnicas de esta descripción para predicción residual avanzada (ARP). Como se muestra en la Figura 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados para ser decodificados en un momento posterior por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 a través de un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluidos ordenadores de sobremesa, ordenadores portátiles (es decir, portátiles), ordenadores de tableta, decodificadores, teléfonos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados a decodificar a través de un medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita 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 por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. El medio de comunicación puede formar parte de una red basada en paquetes, como una red de área local, una red de área amplia o una red global, como Internet. El medio de comunicación puede incluir enrutadores, 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, los datos codificados pueden emitirse desde la interfaz de salida 22 a un dispositivo de almacenamiento. De manera similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede almacenar el vídeo codificado generado por el dispositivo de origen 12.

El dispositivo de destino 14 puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento mediante transmisión en flujo continuo o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de archivos ilustrativos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectado a la red (NAS) o un conjunto de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos convencional, incluida una

conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión en flujo continuo, una transmisión de descarga o una combinación de las mismas.

Las técnicas de esta divulgación no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como retransmisiones televisivas en abierto, retransmisiones de televisión por cable, retransmisiones de televisión por satélite, retransmisiones de vídeo de transmisión por Internet, tales como transmisión dinámica adaptativa 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 admitir la transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión de vídeo, reproducción de vídeo, retransmisión de vídeo y/o videotelefonía.

En el ejemplo de la Figura 1, el dispositivo de origen 12 incluye la fuente de vídeo 18, el codificador de vídeo 20 y la interfaz de salida 22. El dispositivo de destino 14 incluye la interfaz de entrada 28, el decodificador de vídeo 30 y el dispositivo de visualización 32. De acuerdo con esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede configurarse para aplicar las técnicas para la predicción del vector de movimiento en la codificación multi-vista. 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 de una fuente de vídeo externa 18, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema ilustrado 10 de la Figura 1 es solo un ejemplo. Las técnicas para la predicción residual avanzada pueden realizarse mediante cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque generalmente las técnicas de esta divulgación se realizan mediante un dispositivo de codificación de vídeo, las técnicas también se pueden realizar mediante un codificador/descodificador de vídeo, normalmente denominado "CODEC". Además, las técnicas de esta divulgación también pueden realizarse mediante un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación en los que el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de una manera sustancialmente simétrica de manera que cada uno de los dispositivos 12, 14 incluya componentes de codificación y decodificación de vídeo. Por lo tanto, el sistema 10 puede soportar transmisión de vídeo unidireccional o bidireccional entre dispositivos de vídeo 12, 14, por ejemplo, para transmisión de vídeo, reproducción de vídeo, transmisión de vídeo o videotelefonía.

La fuente de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una interfaz de 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 informáticos como el vídeo de origen, o una combinación de vídeo en vivo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una cámara de vídeo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o teléfonos de vídeo. 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 por cable. En cada caso, el vídeo capturado, precapturado o generado por ordenador puede codificarse mediante el codificador de vídeo 20. La información de vídeo codificada puede a continuación enviarse mediante la interfaz de salida 22 a un medio legible por ordenador 16.

El medio legible por ordenador 16 puede incluir medios transitorios, tales como una difusión inalámbrica o transmisión de red por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como un disco duro, conjunto flash, disco compacto, disco de vídeo digital, disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no se muestra) 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, a través de una transmisión de red. De manera similar, un dispositivo informático de una instalación de producción de medios, como una instalación de estampado de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y producir un disco que contenga los datos de vídeo codificados. Por lo tanto, puede entenderse que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de diversas formas y en diversos ejemplos.

La interfaz de entrada 28 del dispositivo de destino 14 recibe información del 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 utilizada por el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o procesamiento de bloques y otros conjuntos codificados, por ejemplo GOP. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de

visualización.

Aunque no se muestra en la Figura 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden integrarse cada uno con un codificador y decodificador de audio, y pueden incluir conjuntos MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación de audio y vídeo en un flujo de datos común o flujos de datos separados. Si corresponde, los conjuntos MUX-DEMUX pueden cumplir con el protocolo multiplexor ITU H.223 u otros protocolos como el protocolo de datagramas de usuario (UDP).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de aplicación (ASIC), matrices de compuertas programables en campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cada uno de los codificadores de vídeo 20 y decodificadores de vídeo 30 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/descodificador de vídeo combinado (CODEC). 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.

Esta descripción puede referirse generalmente al codificador de vídeo 20 "señalizando" cierta información a otro dispositivo, tal como el decodificador de vídeo 30. Debería entenderse, sin embargo, que el codificador de vídeo 20 puede señalar información mediante la asociación de determinados elementos sintácticos con diversas porciones codificadas de datos de vídeo. Es decir, el codificador de vídeo 20 puede "señalizar" datos almacenando ciertos elementos sintácticos en encabezados de varias porciones codificadas de datos de vídeo. En algunos casos, dichos elementos de sintaxis se pueden codificar y almacenar (por ejemplo, almacenar en el dispositivo de almacenamiento 24) antes de ser recibidos y decodificados por el decodificador de vídeo 30. Por lo tanto, el término "señalización" puede referirse generalmente a la comunicación de datos sintácticos u otros datos para decodificar datos de vídeo comprimidos, ya sea que dicha comunicación ocurra en tiempo real o casi real o en un lapso de tiempo, tal como podría ocurrir cuando se almacenan elementos sintácticos en un medio en el momento de la codificación, que luego puede ser recuperado por un dispositivo de decodificación en cualquier momento después de ser almacenado en este medio.

En algunos casos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con estándares patentados o de la industria, tales como el estándar ITU-T H.264, denominado alternativamente MPEG-4, Parte 10, Codificación de Vídeo Avanzada (AVC) o extensiones de dichos estándares. La norma ITU-T H.264/MPEG-4 (AVC) fue formulada por el Grupo de Expertos en Codificación de Vídeo (VCEG) de la ITU-T junto con el Grupo de Expertos en Imágenes en Movimiento (MPEG) de la ISO/IEC como producto de una asociación colectiva conocida como el Equipo Conjunto de Vídeo (JVT).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar adicional o alternativamente de acuerdo con otra norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficiencia (HEVC). Un borrador del estándar HEVC, denominado "Borrador de trabajo HEVC 9" se describe en Bross y col., "High Efficiency Video Coding (HEVC) text specification draft 9," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 11ª reunión, Shanghai, China, octubre de 2012.

Además, como se señaló anteriormente, hay esfuerzos continuos para producir codificación de vídeo escalable, codificación multi-vista y extensiones 3DV para HEVC. Por consiguiente, en algunos casos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden realizar codificación de vídeo multi-vista. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementar una extensión multi-vista de HEVC (denominada MV-HEVC), un códec 3DV completo basado en HEVC de profundidad mejorada (denominado 3D-HEVC) o una extensión de codificación de vídeo escalable de HEVC (denominada SHEVC (HEVC escalable) o HSVC (codificación de vídeo escalable de alta eficiencia).

Las técnicas descritas a continuación se pueden implementar junto con una o más de las extensiones HEVC mencionadas anteriormente. Para 3D-HEVC, se pueden incluir y admitir nuevas herramientas de codificación, incluidas las del nivel de conjunto de codificación/conjunto de predicción, tanto para las vistas de textura como de profundidad. A partir del 21 de noviembre de 2013, el software para 3D-HEVC (es decir, 3D-HTM versión 5.0) se puede descargar desde el siguiente enlace: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-5.0/.

En general, el bucle de compensación de movimiento de HEVC es el mismo que en H.264/AVC. Por ejemplo, la reconstrucción de un fotograma actual/en el bucle de compensación de movimiento puede ser igual a los coeficientes descuantificados r más la predicción temporal P :

$$\hat{I} = r + P.$$

En la fórmula anterior, P indica interpredicción unipredictiva para fotogramas P o interpredicción bipredictiva para

fotogramas B.

Sin embargo, el conjunto de compensación de movimiento en HEVC es diferente de la de las normas de codificación de vídeo anteriores. Por ejemplo, el concepto de un macrobloque en los estándares de codificación de vídeo anteriores no existe en HEVC. Más bien, los macrobloques se reemplazan por una estructura jerárquica flexible basada en un esquema de árbol cuádruple genérico. Dentro de este esquema, se definen tres tipos de bloques, es decir, conjuntos de codificación (CU), conjuntos de predicción (PU) y conjuntos de transformación (TU). Un CU es un conjunto básico de división de regiones. El concepto de un CU es análogo al concepto de macrobloque, pero un CU no está restringido a un tamaño máximo y un CU permite la división recursiva en cuatro CU de igual tamaño para mejorar la adaptabilidad del contenido. Un PU es un conjunto básico de inter/intra predicción. En algunos ejemplos, un PU puede contener múltiples particiones de forma arbitraria en un solo PU para codificar eficazmente patrones de imagen irregulares. Un TU es un conjunto básico de transformación. Los TU de un CU se pueden definir independientemente de los PU del CU. Sin embargo, el tamaño de un TU se limita al CU al que pertenece el TU. Esta separación de la estructura de bloques en tres conceptos diferentes puede permitir que cada uno se optimice de acuerdo con su función, lo que puede dar como resultado una eficiencia de codificación mejorada.

En HEVC y otras especificaciones de codificación de vídeo, una secuencia de vídeo normalmente incluye una serie de imágenes. Las imágenes también pueden denominarse "fotogramas". Una imagen puede incluir tres matrices de muestra, denotadas S_L , S_{Cb} y S_{Cr} . S_L es una matriz bidimensional (es decir, un bloque) de muestras de luminancia. S_{Cb} es una matriz bidimensional de muestras de crominancia Cb. S_{Cr} es una matriz bidimensional de muestras de crominancia Cr. Las muestras de crominancia también pueden denominarse en esta invención muestras de "croma". En otros casos, una imagen puede ser monocromática y solo puede incluir una matriz de muestras de luma.

Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de conjuntos de árbol de codificación (CTU). Cada uno de los CTU puede comprender un bloque de árbol de codificación de muestras de luminancia, dos bloques de árbol de codificación correspondientes de muestras de crominancia y estructuras sintácticas utilizadas para codificar las muestras de los bloques de árbol de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, un CU puede comprender un único bloque de codificación y estructuras sintácticas utilizadas para codificar las muestras del bloque de codificación. Un bloque de árbol de codificación es un bloque $N \times N$ de muestras. Un CTU también puede denominarse "bloque de árbol" o "conjunto de codificación más grande" (LCU). Los CTU de HEVC pueden ser ampliamente análogos a los macrobloques de otros estándares, tales como H.264/AVC. Sin embargo, un CTU no se limita necesariamente a un tamaño particular y puede incluir uno o más CU. Un fragmento puede incluir un número entero de CTU ordenados consecutivamente en un orden de escaneo ráster.

Un fragmento codificado puede comprender un encabezado de fragmento y datos de fragmento. El encabezado de fragmento de un fragmento puede ser una estructura sintáctica que incluye elementos sintácticos que proporcionan información sobre el fragmento. Los datos de fragmento pueden incluir CTU codificados del fragmento.

Esta divulgación puede utilizar el término "conjunto de vídeo" o "bloque de vídeo" o "bloque" para referirse a uno o más bloques de muestra y estructuras sintácticas utilizadas para codificar muestras de uno o más bloques de muestras. Los tipos de ejemplo de conjuntos o bloques de vídeo pueden incluir CTU, CU, PU, conjuntos de transformación (TU), macrobloques, particiones de macrobloques, etc. En algunos contextos, la discusión de los PU se puede intercambiar con la discusión de los macrobloques de las particiones de macrobloques.

Para generar un CTU codificado (es decir, para codificar un CTU) en HEVC, el codificador de vídeo 20 puede realizar recursivamente una partición de árbol cuádruple en los bloques de árbol de codificación de un CTU para dividir los bloques de árbol de codificación en bloques de codificación, de ahí el nombre "conjuntos de árbol de codificación". Un bloque de codificación es un bloque $N \times N$ de muestras. Un CU puede ser un bloque de codificación de muestras de luminancia y dos bloques de codificación correspondientes de muestras de crominancia de una imagen que tiene una matriz de muestras de luminancia, una matriz de muestras Cb y una matriz de muestras Cr, y estructuras sintácticas utilizadas para codificar las muestras de los bloques de codificación. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, un CU puede comprender un único bloque de codificación y estructuras sintácticas utilizadas para codificar las muestras del bloque de codificación.

El codificador de vídeo 20 puede dividir un bloque de codificación de un CU en uno o más bloques de predicción. Un bloque de predicción es un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en el que se aplica la misma predicción. Un conjunto de predicción (PU) de un CU puede ser un bloque de predicción de muestras de luminancia, dos bloques de predicción correspondientes de muestras de crominancia de una imagen y estructuras sintácticas utilizadas para predecir las muestras de bloques de predicción. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, un PU puede comprender un único bloque de predicción y estructuras sintácticas utilizadas para predecir el bloque de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luminancia, Cb y Cr para bloques de predicción de luminancia, Cb y Cr de cada PU del CU. Por lo tanto, en esta divulgación, se puede decir que un CU está dividido en uno o más PU. Para facilitar la explicación, esta divulgación puede referirse al tamaño de un bloque de predicción de un PU como simplemente el tamaño del PU.

El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para un PU. Si el codificador de vídeo 20 utiliza intrapredicción para generar los bloques predictivos de un PU, el codificador de vídeo 20 puede generar los bloques predictivos del PU basándose en muestras decodificadas de la imagen asociada con el PU. En esta descripción, la frase "en función de" puede indicar "en función de, al menos en parte,".

Si el codificador de vídeo 20 utiliza interpredicción para generar los bloques predictivos de un PU, el codificador de vídeo 20 puede generar los bloques predictivos del PU basándose en muestras decodificadas de una o más imágenes distintas de la imagen asociada con el PU. Cuando se usa interpredicción para generar los bloques predictivos de un bloque (por ejemplo, un PU), esta divulgación puede referirse al bloque como "intercodificado" o "interpredicho". La interpredicción puede ser unipredictiva (es decir, unipredicción) o bipredictiva (es decir, bipredicción). Para realizar una unipredicción o bipredicción, el codificador de vídeo 20 puede generar una primera lista de imágenes de referencia (RefPicList0) y una segunda lista de imágenes de referencia (RefPicList1) para una imagen actual. Cada una de las listas de imágenes de referencia puede incluir una o más imágenes de referencia. Después de construir una lista de imágenes de referencia (a saber, RefPicList0 y RefPicList1 si está disponible), se puede usar un índice de referencia a una lista de imágenes de referencia para identificar cualquier imagen de referencia incluida en la lista de imágenes de referencia.

Cuando se utiliza uni-predicción, el codificador de vídeo 20 puede buscar las imágenes de referencia en una o ambas RefPicList0 y RefPicList1 para determinar una ubicación de referencia dentro de una imagen de referencia. Además, cuando se utiliza uni-predicción, el codificador de vídeo 20 puede generar, basándose al menos en parte en muestras correspondientes a la ubicación de referencia, los bloques predictivos para el PU. Además, cuando se utiliza uni-predicción, el codificador de vídeo 20 puede generar un único vector de movimiento que indica un desplazamiento espacial entre un bloque de predicción del PU y la ubicación de referencia. El vector de movimiento puede incluir un componente horizontal que especifica un desplazamiento horizontal entre el bloque de predicción del PU y la ubicación de referencia y puede incluir un componente vertical que especifica un desplazamiento vertical entre el bloque de predicción del PU y la ubicación de referencia.

Cuando se utiliza bi-predicción para codificar un PU, el codificador de vídeo 20 puede determinar una primera ubicación de referencia en una imagen de referencia en RefPicList0 y una segunda ubicación de referencia en una imagen de referencia en RefPicList1. El codificador de vídeo 20 puede generar, basándose al menos en parte en muestras correspondientes a la primera y segunda ubicaciones de referencia, los bloques predictivos para el PU. Además, cuando se utiliza bipredicción para codificar el PU, el codificador de vídeo 20 puede generar un primer vector de movimiento que indica un desplazamiento espacial entre un bloque de predicción del PU y la primera ubicación de referencia y un segundo vector de movimiento que indica un desplazamiento espacial entre el bloque de predicción del PU y la segunda ubicación de referencia.

Si el codificador de vídeo 20 utiliza interpredicción para generar los bloques predictivos de un PU, el codificador de vídeo 20 puede generar los bloques predictivos del PU basándose en muestras decodificadas de una o más imágenes distintas de la imagen asociada con el PU. Por ejemplo, el codificador de vídeo 20 puede realizar interpredicción unipredictiva (es decir, unipredicción) o interpredicción bipredictiva (es decir, bipredicción) en un PU.

En los casos en que el codificador de vídeo 20 realiza una uni-predicción en un PU, el codificador de vídeo 20 puede determinar, basándose en un vector de movimiento del PU, una ubicación de referencia en una imagen de referencia. El codificador de vídeo 20 puede entonces determinar un bloque predictivo para el PU. Cada muestra en el bloque predictivo para el PU puede estar asociada con la ubicación de referencia. En algunos ejemplos, una muestra en un bloque predictivo para un PU puede estar asociada con una ubicación de referencia cuando la muestra está dentro de un bloque de muestras que tiene el mismo tamaño que el PU y cuya esquina superior izquierda es la ubicación de referencia. Cada muestra en el bloque predictivo puede ser una muestra real o interpolada de la imagen de referencia.

En los casos en que las muestras de luma del bloque predictivo se basan en muestras de luma interpoladas de la imagen de referencia, el codificador de vídeo 20 puede generar las muestras de luma interpoladas aplicando un filtro de interpolación de 8 tomas a muestras de luma reales de la imagen de referencia. En los casos en que las muestras de croma del bloque predictivo se basan en muestras de croma interpoladas de la imagen de referencia, el codificador de vídeo 20 puede generar las muestras de croma interpoladas aplicando un filtro de interpolación de 4 tomas a muestras de croma reales de la imagen de referencia. En general, el número de tomas de un filtro indica el número de coeficientes necesarios para representar el filtro matemáticamente. Un filtro con un número de toma más alto es generalmente más complejo que un filtro con un número de toma más bajo.

En los casos en que el codificador de vídeo 20 realiza bi-predicción en un PU, el PU tiene dos vectores de movimiento. El codificador de vídeo 20 puede determinar, basándose en los vectores de movimiento del PU, dos ubicaciones de referencia en dos imágenes de referencia. El codificador de vídeo 20 puede entonces determinar, de la manera descrita anteriormente, bloques de referencia asociados con las dos ubicaciones de referencia. El codificador de vídeo 20 puede entonces determinar un bloque de predicción para el PU. Cada muestra en el bloque de predicción puede ser una media ponderada de muestras correspondientes en los bloques de referencia. La ponderación de las muestras

puede basarse en distancias temporales de las imágenes de referencia de la imagen que contiene el PU.

El codificador de vídeo 20 puede dividir un CU en uno o más PU de acuerdo con varios modos de partición. Por ejemplo, si se usa intrapredicción para generar bloques predictivos para los PU de un CU, el CU puede dividirse de acuerdo con un modo PART_2Nx2N o un modo PART_NxN. En el modo PART_2Nx2N, el CU solo tiene un PU. En el modo PART_NxN, el CU tiene cuatro PU de igual tamaño que tienen bloques de predicción rectangulares. Si se utiliza inter predicción para generar bloques predictivos para los PU de un CU, el CU puede dividirse de acuerdo con el modo PART_2Nx2N, el modo PART_NxN, un modo PART_2NxN, un modo PART_Nx2N, un modo PART_2NxnU, un modo PART_2NxD, un modo PART_nLx2N o un modo PART_nRx2N. En el modo PART_2NxN y el modo PART_Nx2N, el CU se divide en dos PU de igual tamaño que tienen bloques de predicción rectangulares. En cada uno del modo PART_2NxnU, el modo PART_2NxD, el modo PART_nLx2N y el modo PART_nRx2N, el CU se divide en dos PU de tamaño desigual que tienen bloques de predicción rectangulares.

Después de que el codificador de vídeo 20 genera bloques predictivos de luminancia, Cb y Cr para una o más PU de un CU, el codificador de vídeo 20 puede generar un bloque residual de luminancia para el CU. Cada muestra en el bloque residual de luminancia del CU indica una diferencia entre una muestra de luminancia en uno de los bloques de luminancia predictivos del CU y una muestra correspondiente en el bloque de codificación de luminancia original del CU. Además, el codificador de vídeo 20 puede generar un bloque residual Cb para el CU. Cada muestra en el bloque residual Cb del CU puede indicar una diferencia entre una muestra Cb en uno de los bloques Cb predictivos del CU y una muestra correspondiente en el bloque de codificación Cb original del CU. El codificador de vídeo 20 también puede generar un bloque residual de Cr para el CU. Cada muestra en el bloque residual de Cr del CU puede indicar una diferencia entre una muestra de Cr en uno de los bloques de Cr predictivos del CU y una muestra correspondiente en el bloque de codificación de Cr original del CU.

Además, el codificador de vídeo 20 puede utilizar la partición de árbol cuádruple para descomponer los bloques residuales de luma, Cb y Cr de un CU en uno o más bloques de transformación de luma, Cb y Cr. Un bloque de transformación es un bloque rectangular (cuadrado o no cuadrado) de muestras sobre el que se aplica la misma transformación. Un conjunto de transformación (TU) de un CU puede ser un bloque de transformación de muestras de luminancia, dos bloques de transformación correspondientes de muestras de crominancia y estructuras sintácticas utilizadas para transformar las muestras de bloques de transformación. Por lo tanto, cada TU de un CU puede estar asociado con un bloque de transformación de luminancia, un bloque de transformación de Cb y un bloque de transformación de Cr. El bloque de transformación de luminancia asociado con el TU puede ser un sub-bloque del bloque residual de luminancia del CU. El bloque de transformación Cb puede ser un sub-bloque del bloque residual Cb del CU. El bloque de transformación de Cr puede ser un sub-bloque del bloque residual de Cr del CU. En imágenes monocromáticas o imágenes que tienen tres planos de color separados, un TU puede comprender un único bloque de transformación y estructuras sintácticas utilizadas para transformar las muestras del bloque de transformación.

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

Después de generar un bloque de coeficientes (por ejemplo, un bloque de coeficientes de luminancia, un bloque de coeficientes Cb o un bloque de coeficientes Cr), el codificador de vídeo 20 puede cuantificar el bloque de coeficientes. La cuantificación generalmente se refiere a un procedimiento en el que los coeficientes de transformación se cuantifican para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes, proporcionando una compresión adicional. El codificador de vídeo 20 puede cuantificar un bloque de coeficientes asociado con un TU de un CU en función de un valor de parámetro de cuantificación (QP) asociado con el CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados con un CU ajustando el valor de QP asociado con el CU. En algunos ejemplos, el valor de QP asociado con el CU puede estar asociado con la imagen o fragmento actual como un todo. Después de que el codificador de vídeo 20 cuantifica un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos sintácticos que indican los coeficientes de transformación cuantificados. Por ejemplo, el codificador de vídeo 20 puede realizar codificación aritmética binaria adaptativa al contexto (CABAC) en los elementos sintácticos que indican los coeficientes de transformación cuantificados.

El codificador de vídeo 20 puede emitir un flujo de bits que incluye una secuencia de bits que forma una representación de datos de vídeo (es decir, imágenes codificadas y datos asociados). El flujo de bits puede comprender una secuencia de conjuntos de capa de abstracción de red (NAL). Un conjunto NAL es una estructura sintáctica que contiene una indicación del tipo de datos en el conjunto NAL y bytes que contienen esos datos en forma de una carga útil de secuencia de bytes sin procesar (RBSP) intercalada según sea necesario con bits de prevención de emulación. Cada uno de los conjuntos NAL incluye un encabezado de conjunto NAL y encapsula un RBSP. El encabezado de conjunto NAL puede incluir un elemento de sintaxis que indica un código de tipo de conjunto NAL. El código de tipo de conjunto

NAL especificado por el encabezado de conjunto NAL de un conjunto NAL indica el tipo de conjunto NAL. Un RBSP puede ser una estructura sintáctica que contiene un número entero de bytes que está encapsulado dentro de un conjunto NAL. En algunos casos, un RBSP incluye cero bits.

Diferentes tipos de conjuntos NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, diferentes tipos de conjuntos NAL pueden encapsular diferentes RBSP para conjuntos de parámetros de vídeo (VPS), conjuntos de parámetros de secuencia (SPS), conjuntos de parámetros de imagen (PPS), fragmentos codificados, SEI, etc. Los conjuntos NAL que encapsulan RBSP para datos de codificación de vídeo (a diferencia de RBSP para conjuntos de parámetros y mensajes SEI) pueden denominarse conjuntos NAL de capa de codificación de vídeo (VCL).

En HEVC, las SPS pueden contener información que se aplica a todos los fragmentos de una secuencia de vídeo codificada (CVS). En HEVC, una CVS puede comenzar a partir de una imagen de actualización de decodificación instantánea (IDR), o una imagen de acceso de enlace roto (BLA), o una imagen de acceso aleatorio limpio (CRA) que es la primera imagen en el flujo de bits, incluidas todas las imágenes posteriores que no son una imagen IDR o BLA. Es decir, en HEVC, una CVS puede comprender una secuencia de conjuntos de acceso que puede consistir, en orden de decodificación, en un conjunto de acceso CRA que es el primer conjunto de acceso en el flujo de bits, un conjunto de acceso IDR o un conjunto de acceso BLA, seguido de cero o más conjuntos de acceso no IDR y no BLA que incluyen todas los conjuntos de acceso posteriores hasta, pero sin incluir, cualquier conjunto de acceso IDR o BLA posterior.

Un VPS es una estructura sintáctica que comprende elementos sintácticos que se aplican a cero o más CVS completas. Un SPS puede incluir un elemento de sintaxis que identifica una VPS que está activa cuando el SPS está activo. Por lo tanto, los elementos sintácticos de una VPS pueden ser más aplicables en general que los elementos sintácticos de un SPS. Un PPS es una estructura sintáctica que comprende elementos sintácticos que se aplican a cero o más imágenes codificadas. Un PPS puede incluir un elemento de sintaxis que identifica un SPS que está activo cuando el PPS está activo. Un encabezado de fragmento de un fragmento puede incluir un elemento de sintaxis que indica un PPS que está activo cuando se está codificando el fragmento.

El decodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el decodificador de vídeo 30 puede analizar el flujo de bits para obtener elementos sintácticos del flujo de bits. El decodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose, al menos en parte, en los elementos sintácticos obtenidos a partir del flujo de bits. El procedimiento para reconstruir los datos de vídeo puede ser generalmente recíproco al procedimiento realizado por el codificador de vídeo 20. Por ejemplo, el decodificador de vídeo 30 puede usar vectores de movimiento de PU para determinar bloques predictivos para los PU de un CU actual. Además, el decodificador de vídeo 30 puede cuantificar de forma inversa los bloques de coeficientes asociados con los TU del CU actual. El decodificador de vídeo 30 puede realizar transformaciones inversas en los bloques de coeficientes para reconstruir bloques de transformaciones asociados con los TU del CU actual. El decodificador de vídeo 30 puede reconstruir los bloques de codificación del CU actual añadiendo las muestras de los bloques predictivos para los PU del CU actual a las muestras correspondientes de los bloques de transformación de los TU del CU actual. Al reconstruir los bloques de codificación para cada CU de una imagen, el decodificador de vídeo 30 puede reconstruir la imagen.

En algunos ejemplos, el codificador de vídeo 20 puede señalar la información de movimiento de un PU utilizando el modo de fusión o el modo de predicción avanzada de vector de movimiento (AMVP). En otras palabras, en HEVC, hay dos modos para la predicción de parámetros de movimiento, uno es el modo de fusión/omisión y el otro es AMVP. La predicción de movimiento puede comprender la determinación de la información de movimiento de un conjunto de vídeo (por ejemplo, un PU) en función de la información de movimiento de uno o más conjuntos de vídeo. La información de movimiento (es decir, los parámetros de movimiento) de un PU puede incluir uno o más vectores de movimiento del PU, uno o más índices de referencia del PU y uno o más indicadores de dirección de predicción.

Cuando el codificador de vídeo 20 señala la información de movimiento de un PU actual usando el modo de fusión, el codificador de vídeo 20 genera una lista de candidatos a la fusión. En otras palabras, el codificador de vídeo 20 puede realizar un procedimiento de construcción de lista de predictores de vectores de movimiento. La lista de candidatos de fusión incluye un conjunto de candidatos de fusión que indican la información de movimiento de los PU que colindan espacial o temporalmente con el PU actual. Es decir, en el modo de fusión, se construye una lista de candidatos de parámetros de movimiento (por ejemplo, índices de referencia, vectores de movimiento, etc.) donde un candidato puede ser de bloques vecinos espaciales y temporales.

Además, en el modo de fusión, el codificador de vídeo 20 puede seleccionar un candidato de fusión de la lista de candidatos de fusión y puede usar la información de movimiento indicada por el candidato de fusión seleccionado como la información de movimiento del PU actual. El codificador de vídeo 20 puede señalar la posición en la lista de candidatos a la fusión del candidato a la fusión seleccionado. Por ejemplo, el codificador de vídeo 20 puede señalar los parámetros de vector de movimiento seleccionados transmitiendo un índice (es decir, un índice de candidato de fusión) que indica una posición dentro de la lista de candidatos del candidato de fusión seleccionado.

El decodificador de vídeo 30 puede obtener, a partir del flujo de bits, el índice en la lista de candidatos (es decir, el índice de candidatos de fusión). Además, el decodificador de vídeo 30 puede generar la misma lista de candidatos de fusión y puede determinar, en función del índice de candidatos de fusión, el candidato de fusión seleccionado. El decodificador de vídeo 30 puede usar la información de movimiento del candidato a la fusión seleccionado para generar bloques predictivos para el PU actual. Es decir, el decodificador de vídeo 30 puede determinar, en función de, al menos en parte, el índice de lista de candidatos, un candidato seleccionado en la lista de candidatos, donde el candidato seleccionado especifica la información de movimiento (por ejemplo, el vector de movimiento) para el PU actual. De esta manera, en el lado del decodificador, una vez que se decodifica el índice, todos los parámetros de movimiento del bloque correspondiente donde los puntos de índice pueden ser heredados por el PU actual.

El modo de omisión es similar al modo de fusión. En el modo de omisión, el codificador de vídeo 20 y el decodificador de vídeo 30 generan y utilizan una lista de candidatos de fusión de la misma manera que el codificador de vídeo 20 y el decodificador de vídeo 30 utilizan la lista de candidatos de fusión en el modo de fusión. Sin embargo, cuando el codificador de vídeo 20 señala la información de movimiento de un PU actual usando el modo de omisión, el codificador de vídeo 20 no señala ningún dato residual para el PU actual. Por consiguiente, el decodificador de vídeo 30 puede determinar, sin el uso de datos residuales, un bloque de predicción para el PU en función de un bloque de referencia indicado por la información de movimiento de un candidato seleccionado en la lista de candidatos de fusión. Debido a que el modo de omisión tiene el mismo procedimiento de derivación de vectores de movimiento que el modo de fusión, las técnicas descritas en esta invención pueden aplicarse tanto al modo de fusión como al de omisión.

El modo AMVP es similar al modo de fusión en que el codificador de vídeo 20 puede generar una lista de candidatos y puede seleccionar un candidato de la lista de candidatos. Sin embargo, cuando el codificador de vídeo 20 señala la información de movimiento RefPicListX (donde X es 0 o 1) de un PU actual utilizando el modo AMVP, el codificador de vídeo 20 puede señalar una diferencia de vector de movimiento RefPicListX (MVD) para el PU actual y un índice de referencia RefPicListX para el PU actual, además de señalar un indicador de predictor de vector de movimiento RefPicListX (MVP) para el PU actual. El indicador de MVP RefPicListX para el PU actual puede indicar la posición de un candidato de AMVP seleccionado en la lista de candidatos de AMVP. El MVD RefPicListX para el PU actual puede indicar una diferencia entre un vector de movimiento RefPicListX del PU actual y un vector de movimiento del candidato AMVP seleccionado. De esta manera, el codificador de vídeo 20 puede señalar la información de movimiento RefPicListX del PU actual mediante la señalización de un indicador MVP RefPicListX, un valor de índice de referencia RefPicListX y un MVD RefPicListX. En otras palabras, los datos en el flujo de bits que representan el vector de movimiento para el PU actual pueden incluir datos que representan un índice de referencia, un índice a una lista de candidatos y un MVD. Por lo tanto, los vectores de movimiento elegidos pueden señalizarse mediante la transmisión de un índice en la lista de candidatos. Además, también se pueden señalar los valores de índice de referencia y las diferencias del vector de movimiento.

Además, cuando la información de movimiento de un PU actual se señala utilizando el modo AMVP, el decodificador de vídeo 30 puede obtener, a partir del flujo de bits, un MVD para un PU actual y un indicador MVP. El decodificador de vídeo 30 puede generar la misma lista de candidatos de AMVP y puede determinar, en función del indicador de MVP, el candidato de AMVP seleccionado. El decodificador de vídeo 30 puede recuperar un vector de movimiento del PU actual añadiendo el MVD al vector de movimiento indicado por el candidato AMVP seleccionado. Es decir, el decodificador de vídeo 30 puede determinar, en función de un vector de movimiento indicado por el candidato de AMVP seleccionado y el MVD, el vector de movimiento del PU actual. El decodificador de vídeo 30 puede entonces utilizar el vector de movimiento recuperado o los vectores de movimiento del PU actual para generar bloques predictivos para el PU actual.

Cuando un codificador de vídeo genera una lista de candidatos de AMVP para un PU actual, el codificador de vídeo puede derivar uno o más candidatos de AMVP en función de la información de movimiento de los PU que cubren ubicaciones que colindan espacialmente con el PU actual (es decir, PU colindantes espacialmente) y uno o más candidatos de AMVP en función de la información de movimiento de los PU que colindan temporalmente con el PU actual (es decir, PU colindantes temporalmente). En AMVP, se puede derivar una lista candidata de predictores del vector de movimiento para cada hipótesis de movimiento en función de un índice de referencia codificado. En esta divulgación, se puede decir que un PU (u otro tipo de conjunto de vídeo) "cubre" una ubicación si un bloque de predicción asociado con el PU (u otro tipo de bloque de muestra asociado con el conjunto de vídeo) incluye la ubicación. La lista de candidatos incluye vectores de movimiento de bloques vecinos que están asociados con el mismo índice de referencia, así como un predictor de vector de movimiento temporal que se deriva en función de los parámetros de movimiento (es decir, información de movimiento) del bloque vecino del bloque colindante en una imagen de referencia temporal.

Para mejorar aún más la eficiencia de codificación, un codificador de vídeo también puede aplicar predicción de movimiento de inter-vista y/o predicción residual de inter-vista. Con respecto a la predicción de movimiento inter-vista, un codificador de vídeo puede codificar un vector de movimiento asociado con un bloque de una vista con respecto a un vector de movimiento asociado con un bloque de una segunda vista diferente, por ejemplo, utilizando los modos de fusión/omisión o el modo AMVP descrito anteriormente. Del mismo modo, como en la predicción residual inter-vista, un codificador de vídeo puede codificar datos residuales de una vista en relación con el residuo de una segunda

vista diferente. En algunos casos, la predicción residual inter-vista se puede lograr aplicando un procedimiento de predicción residual avanzada (ARP), como se describe con mayor detalle a continuación.

En la predicción residual inter-vista, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar un bloque predictivo para predecir un bloque actual. El bloque predictivo para el bloque actual puede basarse en muestras de una imagen de referencia temporal que están asociadas con una ubicación indicada por un vector de movimiento del bloque actual. La imagen de referencia temporal está asociada con la misma vista que la imagen actual, pero está asociada con una instancia de tiempo diferente a la imagen actual. En algunos casos, cuando las muestras de un bloque se basan en muestras de una imagen particular, las muestras pueden basarse en muestras reales o interpoladas de la imagen particular.

El codificador de vídeo 20 y/o el decodificador de vídeo 30 también determinan un bloque de referencia de disparidad en función de muestras de una imagen de referencia de disparidad que se encuentran en una ubicación indicada por un vector de disparidad del bloque actual. La imagen de referencia de disparidad está asociada con una vista diferente (es decir, una vista de referencia) que la imagen actual, pero está asociada con la misma instancia de tiempo que la imagen actual.

El codificador de vídeo 20 y/o el decodificador de vídeo 30 también determinan un bloque de referencia de disparidad temporal para el bloque actual. El bloque de referencia temporal se basa en muestras de una imagen de referencia de disparidad temporal que están asociadas con una ubicación indicada por el vector de movimiento del bloque actual y el vector de disparidad (por ejemplo, por la combinación del vector de movimiento y el vector de disparidad). Es decir, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden combinar el vector de movimiento y el vector de disparidad y aplicar el vector combinado al bloque actual para localizar el bloque de referencia de disparidad temporal en la imagen de referencia de disparidad temporal. Por lo tanto, la imagen de referencia de disparidad temporal está asociada con la misma vista que la imagen de referencia de disparidad y está asociada con el mismo conjunto de acceso que la imagen de referencia temporal.

El codificador de vídeo 20 y/o el decodificador de vídeo 30 determinan entonces un predictor residual para predecir el residuo asociado con el bloque actual, por ejemplo, la diferencia entre el bloque actual y el bloque de referencia temporal. Cada muestra del predictor residual para el bloque actual indica una diferencia entre una muestra del bloque de referencia de disparidad y una muestra correspondiente del bloque de referencia de disparidad temporal. En algunos casos, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden aplicar un factor de ponderación (por ejemplo, 0, 0,5, 1 o similar) al predictor residual para aumentar la precisión del predictor residual.

El codificador de vídeo 20 puede determinar un bloque residual final para el bloque actual. El bloque residual final comprende muestras que indican diferencias entre muestras del bloque actual, muestras en el bloque predictivo temporal y muestras en el predictor residual. El codificador de vídeo 20 puede incluir, en un flujo de bits, datos que representan el bloque residual final. Decodificador de vídeo, el decodificador de vídeo puede reconstruir el bloque actual en función del bloque residual final (como se obtiene, por ejemplo, a partir de un flujo de bits codificado), el predictor residual y el bloque predictivo temporal.

Si bien ARP puede mejorar la eficiencia de codificación de la predicción residual inter-vista (o inter-capa), son posibles más refinamientos. Por ejemplo, determinadas técnicas de esta divulgación se refieren al factor de ponderación ARP. Como se señaló anteriormente, un codificador de vídeo puede aplicar un factor de ponderación al predictor residual. En general, el factor de ponderación siempre se señala en el flujo de bits, independientemente de si hay una imagen de referencia temporal en las listas de imágenes de referencia para codificar el bloque actual. Sin embargo, la señalización del factor de ponderación cuando no hay una imagen de referencia temporal puede aumentar innecesariamente la complejidad y disminuir la eficiencia, porque si no hay imágenes de referencia temporales, no hay predicción temporal y residuo asociado para aplicar ARP.

De acuerdo con aspectos de esta divulgación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar, para un primer bloque de datos de vídeo en una primera ubicación temporal, si las listas de imágenes de referencia (por ejemplo, RefPicList0 y RefPicList1) para codificar el primer bloque contienen al menos una imagen de referencia en una segunda ubicación temporal diferente. El codificador de vídeo 20 y/o el decodificador de vídeo 30 también pueden codificar el primer bloque de datos de vídeo con respecto a al menos un bloque de referencia de datos de vídeo de una imagen de referencia en las listas de imágenes de referencia. Sin embargo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden deshabilitar un procedimiento de predicción residual de inter-vista cuando las listas de imágenes de referencia no incluyen al menos una imagen de referencia en la segunda ubicación temporal.

El codificador de vídeo 20 puede no señalar un factor de ponderación en el flujo de bits (omitir la señalización de un factor de ponderación), lo que indica que no se utiliza la predicción residual inter-vista. En tales casos, el codificador de vídeo 20 puede codificar el residuo sin predecir el residuo. Del mismo modo, cuando la predicción inter-vista está deshabilitada, el decodificador de vídeo 30 puede determinar automáticamente (es decir, inferir) que un factor de ponderación es igual a cero y omitir la decodificación del factor de ponderación. De esta manera, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden habilitar o deshabilitar la predicción residual inter-vista (por ejemplo,

ARP) en función de las imágenes de referencia en las listas de imágenes de referencia para el bloque que se está codificando actualmente.

Las técnicas descritas anteriormente se pueden aplicar en el contexto de imágenes de acceso aleatorio. Por ejemplo, de acuerdo con los aspectos de esta divulgación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden habilitar o deshabilitar la predicción residual inter-vista en función de si el componente de vista que se codifica actualmente es un componente de vista de acceso aleatorio. Es decir, por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden deshabilitar la predicción residual inter-vista cuando para todos los bloques de una imagen de acceso aleatorio, que no tienen imágenes de referencia temporales asociadas.

Las técnicas de esta descripción también se refieren a la interpolación en la predicción residual inter-vista. Por ejemplo, cuando se realiza la predicción residual inter-vista, tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden usar un procedimiento de compensación de movimiento adicional durante la codificación. Por lo tanto, si un vector de movimiento indica una ubicación de pel fraccional, el codificador de vídeo realiza dos procedimientos de interpolación de pel fraccional, por ejemplo, un procedimiento de interpolación para ubicar el bloque de referencia temporal y otro procedimiento de interpolación para ubicar el bloque de referencia de disparidad temporal. Además, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden aplicar otro procedimiento de interpolación de pel fraccional al determinar un bloque de referencia de disparidad. En HEVC, por ejemplo, se especifica un filtro de 8 tomas para los componentes de luma, mientras que se especifica un filtro de 4 tomas para los componentes de croma. Dichos procedimientos de interpolación pueden aumentar la complejidad informática asociada con la predicción residual inter-vista.

De acuerdo con aspectos de esta descripción, el procedimiento de compensación de movimiento de predicción residual inter-vista puede simplificarse, particularmente con respecto a la interpolación de sub-pel de bloques de referencia. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden interpolar, con un primer tipo de interpolación, una ubicación de un bloque de referencia temporal indicado por un vector de movimiento temporal a un bloque actual de datos de vídeo, donde el bloque actual y el bloque de referencia temporal están ubicados en una primera capa de datos de vídeo. Además, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden interpolar, con un segundo tipo de interpolación, una ubicación de un bloque de referencia de disparidad indicado por un vector de disparidad del bloque actual, donde el bloque de referencia de disparidad está ubicado en una segunda capa diferente, y donde el segundo tipo de interpolación comprende un filtro bilineal. El codificador de vídeo 20 y/o el decodificador de vídeo 30 también pueden determinar un bloque de referencia de disparidad temporal del bloque de referencia de disparidad indicado aplicando el vector de movimiento temporal al bloque de referencia de disparidad, y codificar el bloque actual en función del bloque de referencia temporal, el bloque de referencia de disparidad y el bloque de referencia de disparidad temporal (por ejemplo, codificar el residuo del bloque actual usando predicción residual inter-vista).

Según algunos ejemplos, el primer tipo de interpolación también puede comprender un filtro de paso bajo, tal como un filtro bilineal. En otro ejemplo, se puede utilizar un filtro bilineal para interpolar la ubicación del bloque de referencia de disparidad temporal. Por consiguiente, de acuerdo con aspectos de esta divulgación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden usar un filtro de paso bajo, tal como un filtro bilineal para interbloquear la ubicación de uno o más bloques de referencia en la predicción residual inter-vista. De nuevo, si bien se hace referencia a filtros bilineales, en otros ejemplos, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden aplicar una serie de otros filtros de paso bajo que son más eficientes computacionalmente que la aplicación de los filtros de derivación más altos especificados por HEVC (en particular, los filtros especificados en WD9). De acuerdo con aspectos de esta divulgación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden aplicar los filtros de paso bajo descritos anteriormente a cualquier combinación de componentes de luminancia, componentes de crominancia o componentes de luminancia y crominancia.

Las técnicas de esta divulgación también se refieren a la señalización de un factor de ponderación ARP para modos de codificación y/o modos de partición particulares. Por ejemplo, en general, se puede señalar un factor de ponderación para todos los modos de partición (como se describe con mayor detalle, por ejemplo, con respecto al ejemplo que se muestra en la Figura 12) que incluyen PART_2Nx2N, PART_2NxN, PART_Nx2N o similares, y todos los modos intercodificados que incluyen salto, fusión, predicción avanzada de vectores de movimiento (AMVP). Señalar el factor de ponderación para todos los modos de partición e intermodos puede aumentar innecesariamente la complejidad y disminuir la eficiencia, porque ARP puede no aplicarse de manera eficiente con ciertos modos de partición o intermodos.

De acuerdo con los aspectos de esta descripción, la predicción residual inter-vista puede habilitarse o deshabilitarse en función de un modo de partición y/o modo de codificación del bloque que se está codificando actualmente. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar un modo de partición para codificar un bloque de datos de vídeo, donde el modo de partición indica una división del bloque de datos de vídeo para codificación predictiva. Además, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar si codifica un factor de ponderación para un procedimiento de predicción residual inter-vista en función del modo de partición, donde, cuando el factor de ponderación no está codificado, el procedimiento de predicción residual inter-

vista no se aplica para predecir un residuo para el bloque actual. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden entonces codificar el bloque de datos de vídeo usando el modo de partición determinado.

De acuerdo con aspectos de esta descripción, en algunos ejemplos, el factor de ponderación para cualquier bloque intercodificado con modo de partición desigual a PART_2Nx2N puede no señalizarse. En otro ejemplo, adicional o alternativamente, el factor de ponderación para cualquier bloque intercodificado con modo de codificación desigual al modo de salto y/o modo de fusión puede no señalizarse.

Las técnicas de la presente descripción también se refieren a refinar la manera en que se señalan los factores de ponderación en el flujo de bits. Por ejemplo, en general, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden seleccionar un factor de ponderación de un conjunto fijo de tres factores de ponderación fijos (por ejemplo, 0, 0,5 y 1). Sin embargo, en algunos casos, tres factores de ponderación estáticos pueden no proporcionar suficiente flexibilidad para lograr una eficiencia de predicción suficiente, debido a las diferencias de calidad entre una vista actual y su vista de referencia. Las diferencias de calidad entre la vista actual y la vista de referencia pueden ser dinámicas, en particular con respecto a la codificación de vídeo escalable. Por el contrario, tres factores de ponderación pueden exceder las necesidades de algunos fragmentos o imágenes. Es decir, es posible que algunos fragmentos o imágenes no tengan que seleccionar entre tres factores de ponderación para lograr un equilibrio óptimo entre la complejidad y la mejora de la eficiencia de la codificación.

De acuerdo con los aspectos de esta descripción, se puede implementar un enfoque más flexible para los factores de ponderación. Por ejemplo, la cantidad de factores de ponderación disponibles puede alterarse a nivel de secuencia (por ejemplo, en un conjunto de parámetros, tal como un conjunto de parámetros de secuencia (SPS)). En un ejemplo con fines ilustrativos, un indicador puede señalizarse en un SPS para desactivar uno o más factores de ponderación, por ejemplo, 0,5 y/o 1. En otro ejemplo, dicho indicador puede señalizarse en VPS y aplicarse a todas las vistas sin base. En otro ejemplo más, dicho indicador puede señalizarse en la extensión del conjunto de parámetros de vídeo (VPS) para cada vista sin base. En otro ejemplo, dicho indicador puede proporcionarse en un conjunto de parámetros de imagen (PPS), un encabezado de fragmento o un conjunto de parámetros de vista para deshabilitar uno o más factores de ponderación. Cuando se ha deshabilitado un factor de ponderación, se pueden usar menos bits para representar los factores de ponderación restantes, proporcionando así un ahorro de bits.

Según otros aspectos, se puede proporcionar un indicador para modificar y/o reemplazar uno o más factores de ponderación. En un ejemplo, el codificador de vídeo puede reemplazar el factor de ponderación de 0,5 por un factor de ponderación de 0,75. Este indicador puede señalizarse en un encabezado de fragmento, un SPS, un conjunto de parámetros de imagen (PPS) o un VPS.

Como se señaló anteriormente, un bloque de referencia de disparidad temporal para determinar el predictor residual se localiza típicamente aplicando el vector de movimiento temporal al bloque de referencia de disparidad. Es decir, el codificador de vídeo puede combinar el vector de movimiento temporal y el vector de disparidad y localizar el bloque de referencia de disparidad temporal en función de la combinación, por ejemplo, en relación con el bloque actual. Sin embargo, en algunos casos, el búfer de imágenes decodificadas y/o las listas de imágenes de referencia para codificar el bloque actual pueden no contener la imagen indicada aplicando el vector de movimiento temporal al bloque de referencia de disparidad.

De acuerdo con los aspectos de esta divulgación, un codificador de vídeo puede habilitar o deshabilitar ARP en función de las imágenes de un búfer de imágenes decodificadas y/o listas de imágenes de referencia. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar, para un primer bloque de datos de vídeo en una primera capa de datos de vídeo, un vector de movimiento temporal y una imagen de referencia temporal asociada para predecir el primer bloque, donde la imagen de referencia temporal tiene un valor de recuento de orden de imagen. Además, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar un bloque de referencia de disparidad en una imagen de un conjunto de acceso que incluye una imagen que contiene el primer bloque. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar si un búfer de imágenes decodificadas contiene una imagen de referencia de disparidad temporal que tiene el valor de recuento de orden de imagen de la imagen de referencia temporal, donde la imagen de referencia de disparidad temporal se encuentra basada en una combinación del vector de movimiento temporal y el vector de disparidad, y cuando el búfer de imágenes decodificadas no contiene una imagen de referencia de disparidad temporal que tiene el valor de recuento de orden de imagen de la imagen de referencia temporal, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden modificar el procedimiento de predicción residual inter-vista para predecir datos residuales del primer bloque.

En algunos ejemplos, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden modificar el procedimiento de predicción residual inter-vista deshabilitando el procedimiento de predicción residual inter-vista, de modo que el bloque actual no se codifica usando predicción residual inter-vista. En otros ejemplos, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden modificar el procedimiento de predicción residual inter-vista escalando el vector de movimiento temporal para identificar otra imagen de referencia de disparidad temporal. Por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden escalar el vector de movimiento temporal de modo que, cuando se aplica a la imagen de referencia de disparidad (por ejemplo, o se combina con el vector de disparidad), el vector de

movimiento escalado identifica una imagen de referencia de disparidad temporal que se incluye en la lista de imágenes de referencia y está en una ubicación temporalmente más cercana a la imagen de referencia de disparidad. Las técnicas descritas anteriormente pueden evitar que el codificador de vídeo 20 y/o el decodificador de vídeo 30 intenten localizar el bloque de referencia de disparidad en una imagen que no está incluida en la lista de imágenes de referencia.

La Figura 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar las técnicas descritas en esta invención divulgación. El codificador de vídeo 20 puede realizar una intracodificación e intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una imagen dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en vídeo dentro de imágenes o imágenes adyacentes de una secuencia de vídeo. Intra-modo (modo I) puede referirse a cualquiera de varios modos de compresión basados en el espacio. Los intermodos, como la predicción unidireccional (modo P) o la bi-predicción (modo B), pueden referirse a cualquiera de varios modos de compresión basados en el tiempo.

Como se ha señalado anteriormente, el codificador de vídeo 20 puede adaptarse para realizar una codificación de vídeo multi-vista. Por ejemplo, el codificador de vídeo 20 puede configurarse para codificar múltiples capas escalables de datos de vídeo de acuerdo con el estándar de codificación de vídeo MVC, MV-HEC, 3D-HEVC y/o HSVC. Por lo tanto, el codificador de vídeo 20 puede configurarse para codificar MV-HEVC, de modo que cada vista en una instancia de tiempo puede ser procesada por un decodificador, tal como el decodificador de vídeo 30. Para HEVC-3D, además de codificar mapas de textura (es decir, valores de luma y croma) para cada vista, el codificador de vídeo 20 puede codificar además un mapa de profundidad para cada vista.

En cualquier caso, como se muestra en la Figura 2, el codificador de vídeo 20 recibe datos de vídeo a codificar. En el ejemplo de la Figura 2, el codificador de vídeo 20 incluye un conjunto de selección de modo 40, un sumador 50, un conjunto de procesamiento de transformación 52, un conjunto de cuantificación 54, un conjunto de codificación entrópica 56 y una memoria de imágenes de referencia 64. El conjunto de selección de modo 40, a su vez, incluye el conjunto de compensación de movimiento 42, el conjunto de estimación de movimiento 44, el conjunto de intrapredicción 46 y el conjunto de partición 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye el conjunto de cuantificación inversa 58, el conjunto de transformación inversa 60 y el sumador 62. En algunos ejemplos, un filtro de eliminación de bloques (no mostrado en la Figura 2) también se incluye para filtrar los límites de bloque para eliminar los artefactos de bloques del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtraría típicamente la salida del sumador 62. También se pueden usar filtros en bucle adicionales (en bucle o postbucle) además del filtro de eliminación de bloques. Dichos filtros no se muestran en aras de la brevedad, pero si se desea, pueden filtrar la salida del sumador 50 (como un filtro en bucle).

Durante el procedimiento de codificación, el codificador de vídeo 20 recibe una imagen o fragmento que se va a codificar. La imagen o fragmento se puede dividir en múltiples bloques de vídeo. El conjunto de estimación de movimiento 42 y el conjunto de compensación de movimiento 44 realizan una codificación interpretativa del bloque de vídeo recibido con respecto a uno o más bloques en una o más imágenes de referencia para proporcionar compresión temporal. De manera alternativa, el conjunto de intrapredicción 46 puede realizar una codificación intrapredictiva del bloque de vídeo recibido con respecto a uno o más bloques vecinos en la misma imagen o fragmento que el bloque que se va a codificar para proporcionar compresión espacial. El codificador de vídeo 20 puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.

Además, el conjunto de partición 48 puede dividir bloques de datos de vídeo en sub-bloques, en función de la evaluación de esquemas de partición anteriores en pases de codificación anteriores. Por ejemplo, el conjunto de partición 48 puede dividir inicialmente un fotograma o fragmento en LCU, y dividir cada una de las LCU en sub-CU en función del análisis de tasa-distorsión (por ejemplo, optimización de tasa-distorsión). El conjunto de selección de modo 40 puede producir además una estructura de datos de árbol cuádruple indicativa de la partición de una LCU en sub-CU. Las CU de nodo de hoja del árbol cuádruple pueden incluir una o más PU y una o más TU.

El conjunto de selección de modo 40 puede seleccionar uno de los modos de predicción, intra o inter, por ejemplo, en función de los resultados de error, y proporciona el bloque predicho resultante al sumador 50 para generar datos residuales y al sumador 62 para reconstruir el bloque codificado para su uso como una imagen de referencia. El conjunto de selección de modo 40 también proporciona elementos sintácticos, tales como vectores de movimiento, indicadores intramodo, información de partición y otra información sintáctica similar, a el conjunto de codificación entrópica 56.

El conjunto de estimación de movimiento 42, el conjunto de predicción inter-capas 43 y el conjunto de compensación de movimiento 44 pueden estar altamente integrados, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por el conjunto de estimación de movimiento 42, es el procedimiento de generación de vectores de movimiento, que estiman el movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un PU de un bloque de vídeo dentro de un fotograma o imagen de vídeo actual con respecto a un bloque predictivo dentro de un fotograma de referencia (u otro conjunto codificado) con

respecto al bloque actual que se codifica dentro del fotograma actual (u otro conjunto codificada).

Un bloque predictivo es un bloque que coincide en gran medida con el bloque que se va a codificar, en términos de diferencia de píxeles, que puede determinarse mediante la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxeles subenteros de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64, que también puede denominarse búfer 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 lo tanto, el conjunto de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxel completas y las posiciones de píxel fraccionarias y emitir un vector de movimiento con precisión de píxel fraccionaria.

El conjunto de estimación de movimiento 42 calcula un vector de movimiento para un PU de un bloque de vídeo en un fragmento intercodificado comparando la posición del PU con la posición de un bloque predictivo de una imagen de referencia. Por consiguiente, 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 se puede seleccionar de 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 combinada (Lista c), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64.

El conjunto de estimación de movimiento 42 puede generar y enviar un vector de movimiento que identifica el bloque predictivo de la imagen de referencia a el conjunto de codificación entrópica 56 y el conjunto de compensación de movimiento 44. Es decir, el conjunto 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 ubicar el bloque predictivo dentro de la imagen identificada.

En algunos ejemplos, en lugar de enviar el vector de movimiento real para un PU actual, el conjunto de predicción inter-capa 43 puede predecir el vector de movimiento para reducir aún más la cantidad de datos necesarios para comunicar el vector de movimiento. En este caso, en lugar de codificar y comunicar el vector de movimiento en sí, el conjunto de predicción inter-capa 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 se puede usar con el MVD para definir el vector de movimiento actual, se puede definir mediante un denominado predictor de vector de movimiento (MVP). En general, para ser un MVP válido, el vector de movimiento que se utiliza para la predicción debe apuntar a la misma imagen de referencia que el vector de movimiento que se está codificando actualmente.

El conjunto de predicción inter-capa 43 puede identificar un predictor de vector de movimiento, por ejemplo, para generar un MVD o fusión, en codificación multi-vista. Por ejemplo, el conjunto de predicción inter-capa 43 puede identificar un vector de movimiento de disparidad de un bloque en un componente de vista diferente que un bloque actual para predecir el vector de movimiento para el bloque actual. En otros ejemplos, el conjunto de predicción inter-capa 43 puede identificar un vector de movimiento temporal de un bloque en un componente de vista diferente a un bloque actual para predecir el vector de movimiento para el bloque actual.

De acuerdo con aspectos de esta descripción, el conjunto de predicción inter-capa 43 puede realizar predicción residual inter-capa. Por ejemplo, el conjunto de predicción inter-capa 43 puede codificar datos residuales de una capa con respecto a datos residuales de una segunda capa diferente. En algunos casos, el conjunto de predicción inter-capa 43 puede determinar primero un bloque predictivo para predecir un bloque actual. El bloque predictivo para el bloque actual puede basarse en muestras de una imagen de referencia temporal que están asociadas con una ubicación indicada por un vector de movimiento del bloque actual. La imagen de referencia temporal está asociada con la misma capa que la imagen actual, pero está asociada con una instancia de tiempo diferente a la imagen actual.

El conjunto de predicción inter-capa 43 también determina un bloque de referencia de disparidad en función de muestras de una imagen de referencia de disparidad que se encuentran en una ubicación indicada por un vector de disparidad del bloque actual. La imagen de referencia de disparidad está asociada con una capa diferente (es decir, una capa de referencia) que la imagen actual, pero está asociada con la misma instancia de tiempo que la imagen actual. El conjunto de predicción inter-capa 43 también determina un bloque de referencia de disparidad temporal para el bloque actual. El bloque de referencia temporal se basa en muestras de una imagen de referencia de disparidad temporal que están asociadas con una ubicación indicada por el vector de movimiento del bloque actual y el vector de disparidad (por ejemplo, por la combinación del vector de movimiento y el vector de disparidad). Por lo tanto, la imagen de referencia de disparidad temporal está asociada con la misma vista que la imagen de referencia de disparidad y está asociada con el mismo conjunto de acceso que la imagen de referencia temporal.

El conjunto de predicción inter-capa 43 luego determina un predictor residual para predecir el residuo asociado con el bloque actual, por ejemplo, la diferencia entre el bloque actual y el bloque de referencia temporal. Cada muestra del predictor residual para el bloque actual indica una diferencia entre una muestra del bloque de referencia de disparidad

y una muestra correspondiente del bloque de referencia de disparidad temporal. En algunos casos, el conjunto de predicción inter-capa 43 puede aplicar un factor de ponderación (por ejemplo, 0, 0,5, 1 o similar) al predictor residual para aumentar la precisión del predictor residual.

5 El conjunto de predicción inter-capa 43 puede determinar un bloque residual final para el bloque actual. El bloque residual final comprende muestras que indican diferencias entre muestras del bloque actual, muestras en el bloque predictivo temporal y muestras en el predictor residual. El codificador de vídeo 20 puede incluir, en un flujo de bits, datos que representan el bloque residual final.

10 De acuerdo con los aspectos de esta descripción, el conjunto de predicción inter-capa 43 puede habilitar o deshabilitar la predicción residual inter-vista (incluida la codificación residual de una capa en relación con el residuo de una segunda capa diferente) en función de las imágenes de referencia en las listas de imágenes de referencia para el bloque que se está codificando actualmente. En un ejemplo, el conjunto de predicción inter-capa 43 puede habilitar o deshabilitar la predicción residual inter-vista en función de si las listas de imágenes de referencia (por ejemplo, RefPicList0 y/o RefPicList1) para el bloque que se está codificando actualmente incluyen cualquier imagen de referencia temporal. De acuerdo con los aspectos de esta descripción, si las listas de imágenes de referencia para un bloque interpredicho incluyen solo imágenes de referencia de inter-vista, el conjunto de predicción inter-capa 43 puede deshabilitar el conjunto de predicción inter-capa 43. En algunos ejemplos, el conjunto de predicción inter-capa 43 puede deshabilitar el conjunto de predicción inter-capa 43 para cada bloque de un componente de vista de acceso aleatorio.

En otro ejemplo, cuando la o las listas de imágenes de referencia del bloque de referencia de disparidad no incluye una imagen de referencia en la misma vista que la imagen de referencia de disparidad que tiene el mismo POC que la imagen de referencia temporal, el conjunto de predicción inter-capa 43 puede modificar la predicción residual inter-vista. La determinación de si modificar la predicción residual inter-vista puede basarse en una o ambas listas de imágenes de referencia (por ejemplo, RefPicList0 y/o RefPicList1). Es decir, es decir, dado un índice de lista de imágenes de referencia actual X (siendo X 0 o 1), en un ejemplo, si la lista de imágenes de referencia con un índice de lista igual a X del bloque de referencia de disparidad no incluye una imagen de referencia en la misma vista que la imagen de referencia de disparidad y que tiene el mismo POC que la imagen de referencia temporal del bloque actual, el conjunto de predicción inter-capa 43 puede modificar el procedimiento ARP. En otro ejemplo, si ninguna de las listas de imágenes de referencia (por ejemplo, ni la lista 0 ni la lista 1) del bloque de referencia de disparidad incluye una imagen de referencia en la misma vista que la imagen de referencia de disparidad y que tiene el mismo POC que la imagen de referencia temporal del bloque actual, el conjunto de predicción inter-capa 43 puede modificar el procedimiento ARP.

En algunos casos, el conjunto de predicción inter-capa 43 puede modificar la predicción residual inter-vista al deshabilitar la predicción residual inter-vista. En otros casos, el conjunto de predicción inter-capa 43 puede modificar el procedimiento de predicción residual inter-vista escalando el vector de movimiento temporal para identificar otra imagen de referencia de disparidad temporal. Por ejemplo, el conjunto de predicción inter-capa 43 puede escalar los vectores de movimiento temporales de modo que, cuando se aplica a la imagen de referencia de disparidad, la combinación escalada del vector de movimiento y el vector de disparidad identifica una imagen de referencia de disparidad temporal que se incluye en la lista de imágenes de referencia y está en una ubicación temporalmente más cercana a la imagen de referencia de disparidad.

Si bien se describe con respecto a la lista de imágenes de referencia, el conjunto de predicción inter-capa 43 puede modificar y/o deshabilitar adicional o alternativamente la predicción residual inter-vista si la memoria de imágenes de referencia 64 (es decir, el búfer de imágenes decodificadas) no contiene una imagen en la misma vista que la imagen de referencia de disparidad que tiene el mismo POC que la imagen de referencia temporal.

En otro ejemplo más, de acuerdo con aspectos de esta divulgación, el conjunto de predicción inter-capa 43 puede simplificar la manera en que se ubican los bloques de referencia, particularmente cuando se interpola una posición de sub-pel. Por ejemplo, el conjunto de predicción inter-capa 43 puede usar un filtro de paso bajo, tal como un filtro bilineal, para interpolar la ubicación del bloque de referencia de disparidad. Adicional o alternativamente, el conjunto de predicción inter-capa 43 puede usar un filtro de paso bajo, tal como el filtro bilineal, para interpolar la ubicación del bloque de referencia de disparidad temporal. En incluso otro ejemplo, de acuerdo con aspectos de esta divulgación, el conjunto de estimación de movimiento 42 y/o el conjunto de compensación de movimiento 44 pueden usar un filtro de paso bajo, tal como el filtro bilineal para interpolar la ubicación del bloque de referencia temporal.

En otro ejemplo más, de acuerdo con aspectos de esta divulgación, el conjunto de predicción inter-capa 43 solo puede aplicar la predicción residual inter-vista y, por lo tanto, solo puede señalar un factor de ponderación, para modos de codificación y/o modos de partición particulares. Por ejemplo, el conjunto de predicción inter-capa 43 solo puede señalar un factor de ponderación para cualquier bloque intercodificado con modo de partición desigual a PART_2Nx2N. En otro ejemplo, adicional o alternativamente, el conjunto de predicción inter-capa 43 puede no señalar un factor de ponderación para cualquier bloque intercodificado con un modo de codificación desigual al modo de omisión y/o modo de fusión.

La compensación de movimiento, realizada por el conjunto de compensación de movimiento 44, puede implicar buscar o generar el bloque predictivo basándose en el vector de movimiento determinado por el conjunto de estimación de movimiento 42 y/o la información del conjunto de predicción inter-capas 43. El conjunto de compensación de movimiento 44 puede, en algunos casos, aplicar la predicción inter-vista. Nuevamente, en algunos ejemplos, el conjunto de estimación de movimiento 42, el conjunto de predicción entre capas 43 y el conjunto de compensación de movimiento 44 pueden integrarse funcionalmente. Al recibir el vector de movimiento para el PU del bloque de vídeo actual, el conjunto de compensación de movimiento 44 puede ubicar 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 vídeo residual restando los valores de píxeles del bloque predictivo de los valores de píxeles del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxeles, como se analiza a continuación. En general, el conjunto de estimación de movimiento 42 realiza la estimación de movimiento con respecto a los componentes de luminancia, y el conjunto de compensación de movimiento 44 utiliza vectores de movimiento calculados en función de los componentes de luminancia tanto para los componentes de crominancia como para los componentes de luminancia. El conjunto de selección de modo 40 también puede generar elementos sintácticos asociados con los bloques de vídeo y el fragmento de vídeo para su uso por el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del fragmento de vídeo.

El conjunto de intrapredicción 46 puede intrapredicir un bloque actual, como una alternativa a la interpretación realizada por el conjunto de estimación de movimiento 42 y el conjunto de compensación de movimiento 44, como se ha descrito anteriormente. En particular, el conjunto de intrapredicción 46 puede determinar un modo de intrapredicción para usar para codificar un bloque actual. En algunos ejemplos, el conjunto de intrapredicción 46 puede codificar un bloque actual usando varios modos de intrapredicción, por ejemplo, durante pases de codificación separados, y el conjunto de intrapredicción 46 (o el conjunto de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado para usar a partir de los modos probados.

Por ejemplo, el conjunto de intrapredicción 46 puede calcular valores de tasa-distorsión usando un análisis de tasa-distorsión para los diversos modos de intrapredicción probados, y seleccionar el modo de intrapredicción que tiene las mejores características de tasa-distorsión entre los modos probados. El análisis de tasa-distorsión generalmente determina una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para producir el bloque codificado, así como una tasa de bits (es decir, una cantidad de bits) utilizada para producir el bloque codificado. El conjunto de intrapredicción 46 puede calcular relaciones a partir de las distorsiones y tasas para los diversos bloques codificados para determinar qué modo de intrapredicción exhibe el mejor valor de distorsión nominal para el bloque.

Después de seleccionar un modo de intrapredicción para un bloque, el conjunto de intrapredicción 46 puede proporcionar información indicativa del modo de intrapredicción seleccionado para el bloque a el conjunto de codificación entrópica 56. El conjunto de codificación entrópica 56 puede codificar la información que indica el modo de intrapredicción seleccionado. El codificador de vídeo 20 puede incluir en los datos de configuración de flujo de bits transmitidos, que pueden incluir una pluralidad de tablas de índices de modo de intrapredicción y una pluralidad de tablas de índices de modo de intrapredicción modificadas (también denominadas tablas de mapeo de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índices de modo de intrapredicción y una tabla de índices de modo de intrapredicción modificada para usar para 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 modo 40 del bloque de vídeo original que se está codificando. El sumador 50 representa el componente o componentes que realizan esta operación de resta. El conjunto de procesamiento de transformación 52 aplica una transformación, tal como una transformación de coseno discreta (DCT) o una transformación conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficientes de transformación. El conjunto de procesamiento de transformación 52 puede realizar otras transformaciones que son conceptualmente similares a DCT. Se podrían utilizar transformaciones de ondículas, transformaciones de enteros, transformaciones de sub-bandas, DST u otros tipos de transformaciones en lugar de una DCT. En cualquier caso, el conjunto de procesamiento de transformación 52 aplica la transformación al bloque residual, produciendo un bloque de coeficientes de transformación residuales. La transformación puede convertir la información residual de un dominio de píxeles a un dominio de transformación, tal como un dominio de frecuencia.

El conjunto de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes al conjunto de cuantificación 54. El conjunto de cuantificación 54 cuantifica los coeficientes de transformación para reducir aún más la tasa de bits. El procedimiento de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, el conjunto de cuantificación 54 puede realizar un escaneo de la matriz que incluye los coeficientes de transformación cuantificados. De manera alternativa, el conjunto de codificación entrópica 56 puede realizar el escaneo.

Después de la cuantificación, el conjunto de codificación entrópica 56 codifica por entropía los coeficientes de transformación cuantificados. Por ejemplo, el conjunto de codificación entrópica 56 puede realizar codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación entrópica de partición de intervalo de probabilidad (PIPE) u otra técnica de codificación entrópica.

El conjunto de cuantificación inversa 58 y el conjunto de transformación inversa 60 aplican cuantificación inversa y transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para uso posterior como bloque de referencia. El conjunto de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de la memoria de imágenes de referencia 64. El conjunto de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles subenteros para su uso en la estimación de movimiento.

El sumador 62 añade el bloque residual reconstruido al bloque de predicción con compensación de movimiento producido anteriormente por el conjunto de compensación de movimiento 44 para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede ser utilizado por el conjunto de estimación de movimiento 42 y el conjunto de compensación de movimiento 44 como un bloque de referencia para intercodificar un bloque en un fotograma de vídeo posterior.

La Figura 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 que puede implementar las técnicas descritas en esta invención divulgación para predecir vectores de movimiento en la codificación multi-vista. En el ejemplo de la Figura 3, el decodificador de vídeo 30 incluye un conjunto de decodificación entrópica 80, un conjunto de procesamiento de predicción 81, un conjunto de cuantificación inversa 86, un conjunto de procesamiento de transformación inversa 88, un sumador 90 y una memoria de imágenes de referencia 92. El conjunto de procesamiento de predicción 81 incluye el conjunto de compensación de movimiento 82, el conjunto de predicción inter-capa 83 y el conjunto de intrapredicción 84.

Como se señaló anteriormente, el decodificador de vídeo 30 puede adaptarse para realizar una codificación de vídeo de visualización múltiple. En algunos casos, el decodificador de vídeo 30 puede configurarse para decodificar HEVC multi-vista. Para HEVC-3D, además de decodificar mapas de textura (es decir, valores de luminancia y crominancia) para cada vista, el decodificador de vídeo 30 puede decodificar además un mapa de profundidad para cada vista.

Durante el procedimiento de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados del codificador de vídeo 20. El conjunto de decodificación entrópica 80 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción y otros elementos sintácticos. El conjunto de decodificación entrópica 80 reenvía los vectores de movimiento y otros elementos sintácticos al conjunto de compensación de movimiento 81. El decodificador de vídeo 30 puede recibir los elementos sintácticos a nivel de fragmento de vídeo y/o a nivel de bloque de vídeo.

Por ejemplo, a modo de fondo, el decodificador de vídeo 30 puede recibir datos de vídeo comprimidos que se han comprimido para su transmisión a través de una red en las denominadas "conjuntos de capa de abstracción de red" o conjuntos NAL. Cada conjunto NAL puede incluir un encabezado que identifica un tipo de datos almacenados en el conjunto NAL. Hay dos tipos de datos que se almacenan comúnmente en los conjuntos NAL. El primer tipo de datos almacenados en un conjunto NAL son los datos de la capa de codificación de vídeo (VCL), que incluyen los datos de vídeo comprimidos. El segundo tipo de datos almacenados en un conjunto NAL se conoce como datos no VCL, que incluyen información adicional, como conjuntos de parámetros que definen datos de encabezado comunes a un gran número de conjuntos NAL e información de mejora complementaria (SEI).

Por ejemplo, los conjuntos de parámetros pueden contener la información de encabezado de nivel de secuencia (por ejemplo, en conjuntos de parámetros de secuencia (SPS)) y la información de encabezado 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 información de encabezado, evitando así la necesidad de transmisiones redundantes para la resistencia a errores.

Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), el conjunto de intra predicción 84 del conjunto de procesamiento de predicción 81 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual basándose en un modo de intra predicción señalado y datos de bloques decodificados previamente de la imagen actual. Cuando la imagen se codifica como un fragmento intercodificado (es decir, B, P o GPB), el conjunto de compensación de movimiento 82 del conjunto de procesamiento de predicción 81 produce bloques predictivos para un bloque de vídeo del fragmento de vídeo actual basándose en los vectores de movimiento y otras

sintaxis. elementos recibidos de la unidad de decodificación de entropía 80. Los bloques predictivos pueden producirse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de imágenes de referencia, la lista 0 y la lista 1, utilizando técnicas de construcción predeterminadas basadas en imágenes de referencia almacenadas en la memoria de imágenes de referencia 92.

El conjunto de compensación de movimiento 82 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual mediante el análisis de los vectores de movimiento y otros elementos sintácticos, y utiliza la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, el conjunto de compensación de movimiento 82 utiliza algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intrapredicción o interpredicción) utilizado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, fragmento B, fragmento P o fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para decodificar los bloques de vídeo en el fragmento de vídeo actual. En algunos ejemplos, el conjunto de compensación de movimiento 82 puede recibir cierta información de movimiento desde el conjunto de predicción inter-capa 83.

El conjunto de predicción inter-capa 83 puede recibir datos de predicción que indican dónde recuperar información de movimiento para un bloque actual. Por ejemplo, el conjunto de predicción inter-capa 83 puede recibir información de predicción de vector de movimiento tal como un índice MVP (*mvp_flag*), MVD, bandera de fusión (*merge_flag*) y/o índice de fusión (*merge_idx*) y utilizar dicha información para identificar información de movimiento utilizada para predecir un bloque actual. Es decir, como se señaló anteriormente con respecto al codificador de vídeo 20, de acuerdo con aspectos de esta divulgación, el conjunto de predicción inter-capa 83 puede recibir un índice MVP (*mvp_flag*) y MVD, y usar dicha información para determinar un vector de movimiento utilizado para predecir un bloque actual. El conjunto de predicción inter-capa 83 puede generar una lista de candidatos a MVP o fusión. El MVP y/o los candidatos a la fusión pueden incluir uno o más bloques de vídeo ubicados en una vista diferente a la de un bloque de vídeo que se está decodificando actualmente.

De acuerdo con aspectos de esta descripción, el conjunto de predicción inter-capa 83 puede realizar predicción residual inter-capa. Por ejemplo, el conjunto de predicción inter-capa 83 puede codificar datos residuales de una capa con respecto a datos residuales de una segunda capa diferente. En algunos casos, el conjunto de predicción inter-capa 83 puede determinar primero un bloque predictivo para predecir un bloque actual. El bloque predictivo para el bloque actual puede basarse en muestras de una imagen de referencia temporal que están asociadas con una ubicación indicada por un vector de movimiento del bloque actual. La imagen de referencia temporal está asociada con la misma capa que la imagen actual, pero está asociada con una instancia de tiempo diferente a la imagen actual.

El conjunto de predicción inter-capa 83 también determina un bloque de referencia de disparidad en función de muestras de una imagen de referencia de disparidad que se encuentran en una ubicación indicada por un vector de disparidad del bloque actual. La imagen de referencia de disparidad está asociada con una capa diferente (es decir, una capa de referencia) que la imagen actual, pero está asociada con la misma instancia de tiempo que la imagen actual. El conjunto de predicción inter-capa 83 también determina un bloque de referencia de disparidad temporal para el bloque actual. El bloque de referencia temporal se basa en muestras de una imagen de referencia de disparidad temporal que están asociadas con una ubicación indicada por el vector de movimiento del bloque actual y el vector de disparidad (por ejemplo, por la combinación del vector de movimiento y el vector de disparidad). Por lo tanto, la imagen de referencia de disparidad temporal está asociada con la misma vista que la imagen de referencia de disparidad y está asociada con el mismo conjunto de acceso que la imagen de referencia temporal.

El conjunto de predicción inter-capa 83 luego determina un predictor residual para predecir el residuo asociado con el bloque actual, por ejemplo, la diferencia entre el bloque actual y el bloque de referencia temporal. Cada muestra del predictor residual para el bloque actual indica una diferencia entre una muestra del bloque de referencia de disparidad y una muestra correspondiente del bloque de referencia de disparidad temporal. En algunos casos, el conjunto de predicción inter-capa 83 puede aplicar un factor de ponderación (por ejemplo, 0, 0,5, 1 o similar) al predictor residual para aumentar la precisión del predictor residual.

El conjunto de predicción inter-capa 83 puede obtener, a partir del flujo de bits codificado, datos que indican un bloque residual final para el bloque actual. El conjunto de predicción inter-capa 83 puede reconstruir el bloque actual combinando el bloque residual final, el bloque predictivo temporal y las muestras en el predictor residual.

De acuerdo con los aspectos de esta descripción, el conjunto de predicción inter-capa 83 puede habilitar o deshabilitar la predicción residual inter-vista (incluida la codificación residual de una capa en relación con el residuo de una segunda capa diferente) en función de las imágenes de referencia en las listas de imágenes de referencia para el bloque que se está codificando actualmente. En un ejemplo, el conjunto de predicción inter-capa 83 puede habilitar o deshabilitar la predicción residual inter-vista en función de si la lista de imágenes de referencia para el bloque que se está codificando actualmente incluye cualquier imagen de referencia temporal. De acuerdo con aspectos de esta

divulgación, si las listas de imágenes de referencia para un bloque interpredicho incluyen solo imágenes de referencia inter-vista, el conjunto de predicción inter-capa 83 puede deshabilitar el conjunto de predicción inter-capa 83. En algunos ejemplos, el conjunto de predicción inter-capa 83 puede deshabilitar el conjunto de predicción inter-capa 83 para cada bloque de un componente de vista de acceso aleatorio.

En otro ejemplo, cuando la o las listas de imágenes de referencia del bloque de referencia de disparidad no incluye una imagen de referencia en la misma vista que la imagen de referencia de disparidad que tiene el mismo POC que la imagen de referencia temporal, el conjunto de predicción inter-capa 83 puede modificar la predicción residual inter-vista. La determinación de si modificar la predicción residual inter-vista puede basarse en una o ambas listas de imágenes de referencia (por ejemplo, RefPicList0 y/o RefPicList1). Es decir, es decir, dado un índice de lista de imágenes de referencia actual X (siendo X 0 o 1), en un ejemplo, si la lista de imágenes de referencia con un índice de lista igual a X del bloque de referencia de disparidad no incluye una imagen de referencia en la misma vista que la imagen de referencia de disparidad y que tiene el mismo POC que la imagen de referencia temporal del bloque actual, el conjunto de predicción inter-capa 83 puede modificar el procedimiento ARP. En otro ejemplo, si ninguna de las listas de imágenes de referencia (por ejemplo, ni la lista 0 ni la lista 1) del bloque de referencia de disparidad incluye una imagen de referencia en la misma vista que la imagen de referencia de disparidad y que tiene el mismo POC que la imagen de referencia temporal del bloque actual, el conjunto de predicción inter-capa 83 puede modificar el procedimiento ARP.

En algunos casos, el conjunto de predicción inter-capa 83 puede modificar la predicción residual inter-vista al deshabilitar la predicción residual inter-vista. En otros casos, el conjunto de predicción inter-capa 83 puede modificar el procedimiento de predicción residual inter-vista escalando el vector de movimiento temporal para identificar otra imagen de referencia de disparidad temporal. Por ejemplo, el conjunto de predicción inter-capa 83 puede escalar los vectores de movimiento temporales de modo que, cuando se aplica a la imagen de referencia de disparidad, la combinación escalada del vector de movimiento y el vector de disparidad identifica una imagen de referencia de disparidad temporal que se incluye en la lista de imágenes de referencia y está en una ubicación temporalmente más cercana a la imagen de referencia de disparidad.

En otro ejemplo más, de acuerdo con aspectos de esta divulgación, el conjunto de predicción inter-capa 83 puede simplificar la manera en que se ubican los bloques de referencia, particularmente cuando se interpola una posición de sub-pel. Por ejemplo, el conjunto de predicción inter-capa 83 puede usar un filtro de paso bajo, tal como un filtro bilineal, para interpolar la ubicación del bloque de referencia de disparidad. Adicional o alternativamente, el conjunto de predicción inter-capa 83 puede usar un filtro de paso bajo, tal como el filtro bilineal, para interpolar la ubicación del bloque de referencia de disparidad temporal. En incluso otro ejemplo, de acuerdo con aspectos de la presente divulgación, el conjunto de compensación de movimiento 82 puede usar un filtro de paso bajo, tal como el filtro bilineal, para interpolar la ubicación del bloque de referencia temporal.

En otro ejemplo más, de acuerdo con aspectos de esta divulgación, el conjunto de predicción inter-capa 83 solo puede aplicar la predicción residual inter-vista y, por lo tanto, solo puede señalar un factor de ponderación, para modos de codificación y/o modos de partición particulares. Por ejemplo, el conjunto de predicción inter-capa 83 solo puede señalar un factor de ponderación para cualquier bloque intercodificado con modo de partición desigual a PART_2Nx2N. En otro ejemplo, adicional o alternativamente, el conjunto de predicción inter-capa 83 puede no señalar un factor de ponderación para cualquier bloque intercodificado con un modo de codificación desigual al modo de omisión y/o modo de fusión.

El conjunto de cuantificación inversa 86 cuantifica inversamente, es decir, descuantifica, los coeficientes de transformación cuantificados proporcionados en el flujo de bits y decodificados por el conjunto de decodificación entrópica 80. El procedimiento de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QPY calculado por el decodificador de vídeo 20 para cada bloque de vídeo en el fragmento de vídeo para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debe aplicarse.

El conjunto de transformación inversa 88 aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación entera inversa o un procedimiento de transformación inversa conceptualmente similar, a los coeficientes de transformación para producir bloques residuales en el dominio de píxeles. De acuerdo con los aspectos de la presente descripción, el conjunto de procesamiento de transformación inversa 88 puede determinar la manera en que se aplicaron las transformaciones a los datos residuales. Es decir, por ejemplo, el conjunto de procesamiento de transformación inversa 88 puede determinar una RQT que representa la manera en que se aplicaron transformaciones (por ejemplo, DCT, transformación entera, transformación de ondícula o una o más de otras transformaciones) a las muestras de luma residual y las muestras de croma residual asociadas con un bloque de datos de vídeo recibidos.

Después de que el conjunto de compensación de movimiento 82 genera el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales del conjunto de procesamiento de transformación inversa 88 con los bloques predictivos correspondientes generados por el conjunto de compensación de movimiento

82. El sumador 90 representa el componente o componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques decodificados con el fin de eliminar los artefactos de bloques. También se pueden usar otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar la calidad del vídeo. Los bloques de vídeo decodificados en una imagen dada se almacenan luego en la memoria de imágenes de referencia 92, que almacena imágenes de referencia utilizadas para la compensación de movimiento posterior. La memoria de imágenes de referencia 92 también almacena vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la Figura 1.

La Figura 4 es un diagrama conceptual que ilustra un orden de decodificación multi-vista de ejemplo. El orden de decodificación multi-vista puede ser un orden de flujo de bits. En el ejemplo de la Figura 4, cada cuadrado corresponde a un componente de vista. Las columnas de cuadrados corresponden a conjuntos de acceso. Cada conjunto de acceso se puede definir para contener las imágenes codificadas de todas las vistas de una instancia de tiempo. Las filas de cuadrados corresponden a vistas. En el ejemplo de la Figura 4, los conjuntos de acceso se etiquetan T0...T11 y las vistas se etiquetan S0...S7. Debido a que cada componente de vista de un conjunto de acceso se decodifica antes que cualquier componente de vista de la siguiente conjunto de acceso, el orden de decodificación de la Figura 4 puede denominarse como codificación de tiempo primero. El orden de decodificación de los conjuntos de acceso puede no ser idéntico al orden de salida o visualización.

La codificación multi-vista puede admitir la predicción inter-vista. La predicción inter-vista es similar a la predicción inter-vista utilizada en H.264/AVC, HEVC u otras especificaciones de codificación de vídeo y puede utilizar los mismos elementos sintácticos. Sin embargo, cuando un codificador de vídeo realiza la predicción inter-vista en un conjunto de vídeo actual (tal como un macrobloque o PU), el codificador de vídeo puede usar, como una imagen de referencia, una imagen que está en el mismo conjunto de acceso que el conjunto de vídeo actual, pero en una vista diferente. Por el contrario, la inter predicción convencional solo utiliza imágenes en diferentes conjuntos de acceso como imágenes de referencia.

La Figura 5 es un diagrama conceptual que ilustra un patrón de predicción de MVC de ejemplo que se puede usar con MVC, HEVC multi-vista y 3D-HEVC (multi-vista más profundidad). Las referencias a MVC a continuación se aplican a MVC en general, y no se limitan a H.264/MVC.

En el ejemplo de la Figura 5, se ilustran ocho vistas (S0-S7) y se ilustran doce ubicaciones temporales (T0-T11) para cada vista. En general, cada fila en la Figura 5 corresponde a una vista, mientras que cada columna indica una ubicación temporal. Cada una de las vistas puede identificarse usando un identificador de vista ("view_id"), que puede usarse para indicar una ubicación de cámara relativa con respecto a las otras vistas. En el ejemplo que se muestra en la Figura 5, los ID de vista se indican como "S0" a "S7", aunque también se pueden usar ID de vista numéricos. Además, cada una de las ubicaciones temporales puede identificarse usando un valor de recuento de orden de imágenes (POC), que indica un orden de visualización de las imágenes. En el ejemplo que se muestra en la Figura 5, los valores de POC se indican como "T0" a "T11".

Aunque un flujo de bits codificado multi-vista puede tener una llamada vista base que es decodificable por decodificadores particulares y se puede admitir un par de vistas estéreo, algunos flujos de bits multi-vista pueden admitir más de dos vistas como una entrada de vídeo 3D. Por consiguiente, un renderizador de un cliente que tiene un decodificador particular puede esperar contenido de vídeo 3D con múltiples vistas.

Las imágenes en la Figura 5 se indican usando un bloque sombreado que incluye una letra, que designa si la imagen correspondiente está intracodificada (es decir, un fotograma I) o intercodificada en una dirección (es decir, como un fotograma P) o en múltiples direcciones (es decir, como un fotograma B). En general, las predicciones se indican mediante flechas, donde la imagen apuntada utiliza el punto del objeto para la referencia de predicción. Por ejemplo, el fotograma P de la vista S2 en la ubicación temporal T0 se predice a partir del fotograma 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 multi-vista se pueden codificar de forma predictiva con respecto a las imágenes en diferentes ubicaciones temporales. Por ejemplo, el fotograma b de la vista S0 en la ubicación temporal T1 tiene una flecha que lo señala desde el fotograma I de la vista S0 en la ubicación temporal T0, lo que indica que el fotograma b se predice a partir del fotograma I. Además, sin embargo, en el contexto de la codificación de vídeo multi-vista, las imágenes pueden predecirse inter-vista. Es decir, un componente de vista puede usar los componentes de vista en otras vistas como referencia. Por ejemplo, la predicción inter-vista puede realizarse como si el componente de vista en otra vista fuera una referencia de interpredicción. Las posibles referencias inter-vista pueden señalizarse en la extensión MVC del conjunto de parámetros de secuencia (SPS) y pueden modificarse mediante el procedimiento de construcción de la lista de imágenes de referencia, lo que permite un ordenamiento flexible de las referencias de predicción inter-vista o de predicción inter-vista.

La Figura 5 proporciona varios ejemplos de predicción inter-vista. Las imágenes de la vista S1, en el ejemplo de la

Figura 5, se ilustran como predichas a partir de imágenes en diferentes ubicaciones temporales de la vista S1, así como la inter-vista predicha a partir de imágenes de imágenes de las vistas S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, el fotograma b de la vista S1 en la ubicación temporal T1 se predice a partir de cada uno de los fotogramas B de la vista S1 en las ubicaciones temporales T0 y T2, así como los fotogramas b de las vistas S0 y S2 en la ubicación temporal T1.

En el ejemplo de la Figura 5, la "B" mayúscula y la "b" minúscula pretenden indicar diferentes relaciones jerárquicas entre imágenes, en lugar de diferentes metodologías de codificación. En general, los fotogramas "B" mayúscula son relativamente más altos en la jerarquía de predicción que los fotogramas "b" minúscula. La Figura 5 también ilustra variaciones en la jerarquía de predicción utilizando diferentes niveles de sombreado, donde una mayor cantidad de imágenes de sombreado (es decir, relativamente más oscuras) son más altas en la jerarquía de predicción que aquellas imágenes que tienen menos sombreado (es decir, relativamente más claras). Por ejemplo, todos los fotogramas I en la Figura 5 se ilustran con sombreado completo, mientras que los fotogramas P tienen un sombreado algo más claro, y los fotogramas B (y los fotogramas b en minúscula) tienen varios niveles de sombreado entre sí, pero siempre más claro que el sombreado de los fotogramas P y los fotogramas I.

En general, la jerarquía de predicción está relacionada con los índices de orden de visualización, ya que las imágenes relativamente más altas en la jerarquía de predicción deben decodificarse antes de decodificar imágenes que son relativamente más bajas en la jerarquía, de modo que esas imágenes relativamente más altas en la jerarquía pueden usarse como imágenes de referencia durante la decodificación de las imágenes relativamente más bajas en la jerarquía. Un índice de orden de vista es un índice que indica el orden de decodificación de los componentes de vista en un conjunto de acceso. Los índices de orden de vista pueden estar implícitos en un conjunto de parámetros, como un SPS.

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

Se puede extraer un subconjunto de un flujo de bits completo para formar un subflujo de bits conforme. Hay muchos subflujos de bits posibles que pueden requerir aplicaciones específicas, en función de, por ejemplo, un servicio proporcionado por un servidor, la capacidad, el soporte y las capacidades de los decodificadores de uno o más clientes, y/o la preferencia de uno o más clientes. Por ejemplo, un cliente puede requerir solo tres vistas, y puede haber dos escenarios. En un ejemplo, un cliente puede requerir una experiencia de visualización fluida y puede preferir vistas con los valores de `view_id` S0, S1 y S2, mientras que otro cliente puede requerir una escalabilidad de vista y preferir vistas con los valores de `view_id` S0, S2 y S4. Tenga en cuenta que ambos subflujos de bits se pueden decodificar como flujos de bits independientes y se pueden admitir simultáneamente.

Con respecto a la predicción inter-vista, se permite la predicción inter-vista entre imágenes en el mismo conjunto de acceso (es decir, con la misma instancia de tiempo). Al codificar una imagen en una de las vistas sin base, se puede añadir una imagen a una lista de imágenes de referencia, si está en una vista diferente pero con una misma instancia de tiempo. Una imagen de referencia de predicción inter-vista se puede poner en cualquier posición de una lista de imágenes de referencia, al igual que cualquier imagen de referencia de predicción inter.

Por lo tanto, en el contexto de la codificación de vídeo multi-vista, hay dos tipos de vectores de movimiento. Un tipo de vector de movimiento es un vector de movimiento normal que apunta a una imagen de referencia temporal. El tipo de interpredicción correspondiente a un vector de movimiento temporal normal puede denominarse predicción compensada por movimiento (MCP). Cuando se utiliza una imagen de referencia de predicción inter-vista para la compensación de movimiento, el vector de movimiento correspondiente se denomina "vector de movimiento de disparidad". En otras palabras, un vector de movimiento de disparidad apunta a una imagen en una vista diferente (es decir, una imagen de referencia de disparidad o una imagen de referencia inter-vista). El tipo de inter predicción correspondiente a un vector de movimiento de disparidad puede denominarse "predicción compensada por disparidad" o "DCP".

Como se mencionó anteriormente, se está desarrollando una extensión multi-vista de HEVC (es decir, MV-HEVC) y una extensión 3DV de HEVC (es decir, 3D-HEVC). MV-HEVC y 3D-HEVC pueden mejorar la eficiencia de codificación utilizando la predicción de movimiento inter-vista y la predicción residual inter-vista. En la predicción de movimiento inter-vista, un codificador de vídeo puede determinar (es decir, predecir) la información de movimiento de un PU actual basándose en la información de movimiento de un PU en una vista diferente a el PU actual. En la predicción residual inter-vista, un codificador de vídeo puede determinar los bloques residuales de un CU actual en función de los datos residuales en una vista diferente al CU actual mediante el uso de la estructura de predicción que se muestra en la Figura 5.

Para habilitar la predicción de movimiento inter-vista y la predicción residual inter-vista, un codificador de vídeo puede determinar vectores de disparidad para bloques (por ejemplo, PU, CU, etc.). En general, se utiliza un vector de disparidad como estimador del desplazamiento entre dos vistas. Un codificador de vídeo, tal como el codificador de vídeo 20 o el decodificador de vídeo 30, puede usar un vector de disparidad para un bloque para localizar un bloque de referencia (que puede denominarse en esta invención como un bloque de referencia de disparidad) en otra vista para movimiento inter-vista o predicción residual, o el codificador de vídeo puede convertir el vector de disparidad en un vector de movimiento de disparidad para predicción de movimiento inter-vista.

La Figura 6 es un diagrama conceptual que ilustra la codificación de vídeo escalable. Mientras que la Figura 6 se describe con respecto a H.264/AVC y SVC, debe entenderse que pueden codificarse capas similares demandando otros esquemas de codificación de vídeo multicapa, incluyendo HSVC. En otro ejemplo, se pueden codificar capas similares usando un códec de múltiples estándares. Por ejemplo, una capa base puede codificarse usando H.264/AVC, mientras que una capa de mejora puede codificarse usando una extensión escalable solo de HLS para HEVC. Por lo tanto, las referencias a SVC a continuación pueden aplicarse a la codificación de vídeo escalable en general, y no se limitan a H.264/SVC.

En SVC, las escalabilidades se pueden habilitar en tres dimensiones que incluyen, por ejemplo, espacial, temporal y de calidad (representada como una tasa de bits o relación señal a ruido (SNR)). En general, normalmente se puede lograr una mejor representación añadiendo a una representación en cualquier dimensión. Más específicamente, en el ejemplo de la Figura 6, la capa 0 está codificada en formato intermedio común de un cuarto (QCIF) que tiene una velocidad de fotogramas de 7,5 Hz y una velocidad de bits de 64 kilobytes por segundo (KBPS). Además, la capa 1 está codificada en QCIF con una velocidad de fotogramas de 15 Hz y una velocidad de bits de 64 KBPS, la capa 2 está codificada en CIF con una velocidad de fotogramas de 15 Hz y una velocidad de bits de 256 KBPS, la capa 3 está codificada en QCIF con una velocidad de fotogramas de 7.5 Hz y una velocidad de bits de 512 KBPS, y la capa 4 está codificada en 4CIF con una velocidad de fotogramas de 30 Hz y una velocidad de bits de Megabyte por segundo (MBPS). Debe entenderse que el número, el contenido y la disposición particulares de las capas que se muestran en la Figura 5 se proporcionan solo con fines de ejemplo.

En cualquier caso, una vez que un codificador de vídeo (tal como el codificador de vídeo 20) ha codificado contenido de tal manera escalable, un decodificador de vídeo (tal como el decodificador de vídeo 30) puede usar una herramienta extractora para adaptar el contenido entregado real de acuerdo con los requisitos de la aplicación, que pueden depender, por ejemplo, del cliente o del canal de transmisión.

En SVC, las imágenes que tienen la capa espacial y de calidad más baja suelen ser compatibles con H.264/AVC. En el ejemplo de la Figura 6, las imágenes con la capa espacial y de calidad más baja (imágenes en la capa 0 y la capa 1, con resolución QCIF) pueden ser compatibles con H.264/AVC. Entre ellos, las imágenes del nivel temporal más bajo forman la capa base temporal (capa 0). Esta capa base temporal (capa 0) se puede mejorar con imágenes de niveles temporales más altos (capa 1).

Además de la capa compatible con H.264/AVC, se pueden añadir varias capas de mejora espacial y/o de calidad para proporcionar escalabilidades espaciales y/o de calidad. Cada capa de mejora espacial o de calidad en sí misma puede ser escalable temporalmente, con la misma estructura de escalabilidad temporal que la capa compatible con H.264/AVC.

Si bien la predicción residual inter-vista se puede describir con respecto a "vistas" de datos de vídeo, debe entenderse que se pueden aplicar técnicas similares a múltiples capas de datos, tales como capas de la estructura escalable mostrada en la Figura 6. Por ejemplo, un codificador de vídeo (tal como el codificador de vídeo 20 y/o el decodificador de vídeo 30) puede predecir el residuo de una capa usando otra capa. En algún caso, las técnicas pueden implementarse con una extensión escalable de HEVC, tal como HSVC.

En particular, como se describe con mayor detalle a continuación, el codificador de vídeo 20 puede señalar factores de ponderación para CU solo para ciertos modos de partición de codificación y/o para ciertos modos de codificación. Cuando no se señala un factor de ponderación, el decodificador de vídeo 30 puede omitir la decodificación de un factor de ponderación y determinar automáticamente (es decir, inferir) que el factor de ponderación es cero.

En un ejemplo, el factor de ponderación para un CU intercodificada con modo de partición desigual a PART_2Nx2N puede no ser señalado. En un ejemplo alternativo, el factor de ponderación para un CU intercodificada con modo de partición desigual a PART_2Nx2N, PART_2NxN y PART_Nx2N puede no ser señalado. En otro ejemplo más, adicional o alternativamente, el factor de ponderación para cualquier CU intercodificada con modo de codificación desigual para saltar y/o fusionar puede no señalizarse.

Según otros aspectos, el codificador de vídeo puede modificar los factores de ponderación. Por ejemplo, un indicador puede señalizarse en un nivel de secuencia para desactivar uno o más factores de ponderación (por ejemplo, 0,5 y/o 1). En algunos ejemplos, el indicador puede señalizarse en una extensión de VPS para cada vista sin base. En otros ejemplos, el indicador puede señalizarse en un VPS y puede ser aplicable para todas las vistas sin base. En aún otros

ejemplos, el indicador puede señalizarse en un conjunto de parámetros de imagen (PPS), encabezado de fragmento o conjunto de parámetros de visualización.

En otro ejemplo, se puede indicar a un indicador que modifique uno o más de los factores de ponderación. Por ejemplo, el indicador puede hacer que el decodificador de vídeo 30 reemplace un factor de ponderación inicial (por ejemplo, 0,5) con un nuevo factor de ponderación (por ejemplo, 0,75). Este indicador de modificación puede señalizarse en un PPS, un encabezado de fragmento o un VPS.

De acuerdo con aún otros aspectos, el codificador de vídeo puede habilitar o deshabilitar ARP en función de las imágenes de un búfer de imágenes decodificadas y/o listas de imágenes de referencia para codificar una imagen en la estructura escalable mostrada en la Figura 6. Por ejemplo, cuando un búfer de imágenes decodificadas para codificar un PU actual no incluye una imagen en la misma vista que la imagen de referencia de disparidad que tiene el mismo POC que la imagen de referencia temporal, un codificador de vídeo puede modificar el procedimiento ARP para el PU.

En otro ejemplo, adicional/alternativamente, cuando una o ambas de las listas de imágenes de referencia del bloque de referencia de disparidad no incluyen una imagen de referencia en la misma vista que la imagen de referencia de disparidad que tiene el mismo POC que la imagen de referencia temporal, un codificador de vídeo puede modificar el procedimiento ARP para el PU.

En algunos ejemplos, el codificador de vídeo puede modificar el procedimiento ARP deshabilitando el procedimiento ARP, de modo que el PU actual no esté codificada usando ARP. En otros ejemplos, el codificador de vídeo puede modificar el procedimiento ARP escalando el vector de movimiento temporal para identificar otra imagen de referencia de disparidad temporal disponible.

La Figura 7 es un diagrama conceptual que ilustra PU vecinos espacialmente de ejemplo con respecto a un PU 100 actual, que se puede usar para determinar un vector de disparidad para el PU 100 actual. En el ejemplo de la Figura 7, los PU espacialmente vecinos pueden ser PU que cubren las ubicaciones indicadas como A_0 , A_1 , B_0 , B_1 y B_2 .

Como se señaló anteriormente, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede realizar una predicción de movimiento inter-vista y/o una predicción residual inter-vista. Para habilitar estas dos herramientas de codificación, el primer paso es derivar un vector de disparidad.

En algunos ejemplos, el codificador de vídeo puede usar el procedimiento del Vector de Disparidad Basado en Bloques Vecinos (NBDV) para derivar un vector de disparidad para un bloque. Por ejemplo, para derivar un vector de disparidad para PU, se puede usar un procedimiento llamado NBDV en un modelo de prueba para 3D-HEVC (es decir, 3D-HTM). El procedimiento NBDV utiliza vectores de movimiento de disparidad de bloques vecinos espaciales y temporales (como los PU vecinos A_0 , A_1 , B_0 , B_1 y B_2) para derivar el vector de disparidad para un bloque actual. Debido a que es probable que los bloques vecinos (por ejemplo, bloques que colindan espacial o temporalmente con el bloque actual) compartan casi la misma información de movimiento y disparidad en la codificación de vídeo, el bloque actual puede usar la información del vector de movimiento en los bloques vecinos como predictores del vector de disparidad del bloque actual.

Cuando un codificador de vídeo realiza el procedimiento NBDV, el codificador de vídeo puede comprobar, en un orden de comprobación fijo, los vectores de movimiento de los bloques espacialmente vecinos y temporalmente vecinos. Cuando el codificador de vídeo comprueba el o los vectores de movimiento de un bloque espacialmente vecino o temporalmente vecino, el codificador de vídeo puede determinar si el o los vectores de movimiento del bloque espacialmente vecino o temporalmente vecino son vectores de movimiento de disparidad. Un vector de movimiento de disparidad de un bloque de una imagen es un vector de movimiento que apunta a una ubicación dentro de una imagen de referencia de disparidad de la imagen.

Una imagen de referencia de disparidad de una imagen dada puede ser una imagen que está asociada con el mismo conjunto de acceso que la imagen dada, pero está asociada con una vista diferente a la imagen dada. Cuando el codificador de vídeo identifica un vector de movimiento de disparidad, el codificador de vídeo puede finalizar el procedimiento de comprobación. El codificador de vídeo puede convertir el vector de movimiento de disparidad devuelto en un vector de disparidad y puede usar el vector de disparidad para la predicción de movimiento inter-vista y la predicción residual de inter-vistas. Por ejemplo, el codificador de vídeo puede establecer un componente horizontal del vector de disparidad para el bloque actual igual a un componente horizontal del vector de movimiento de disparidad y puede establecer el componente vertical del vector de disparidad en 0.

Si el codificador de vídeo no puede derivar un vector de disparidad para el bloque actual (es decir, si no se encuentra ningún vector de disparidad) realizando el procedimiento NBDV, el codificador de vídeo puede usar un vector de disparidad cero como el vector de disparidad para el bloque actual. El vector de disparidad cero es un vector de disparidad que tiene componentes horizontales y verticales iguales a 0. Por lo tanto, incluso cuando el procedimiento NBDV devuelve un resultado no disponible, otros procedimientos de codificación del codificador de vídeo que

requieren un vector de disparidad pueden usar un vector de disparidad cero para el bloque actual.

En algunos ejemplos, si el codificador de vídeo no puede derivar un vector de disparidad para el bloque actual realizando el procedimiento NBDV, el codificador de vídeo puede deshabilitar la predicción residual inter-vista para el bloque actual. Sin embargo, independientemente de si el codificador de vídeo es capaz de derivar un vector de disparidad para el bloque actual mediante la realización del procedimiento NBDV, el codificador de vídeo puede utilizar la predicción de movimiento inter-vista para el PU actual. Es decir, si no se encuentra ningún vector de disparidad después de verificar todos los bloques vecinos predefinidos, se puede usar un vector de disparidad cero para la predicción de movimiento inter-vista, mientras que la predicción residual de inter-vista se puede deshabilitar para el CU correspondiente.

Como se señaló anteriormente, se pueden usar cinco bloques vecinos espaciales para la derivación del vector de disparidad, que incluyen, por ejemplo, los PU indicadas por A_0 , A_1 , B_0 , B_1 o B_2 . Además, se pueden usar uno o más bloques vecinos temporales para la derivación del vector de disparidad. En este caso, todas las imágenes de referencia de la vista actual se tratan como imágenes candidatas. El número de imágenes candidatas puede limitarse adicionalmente a, por ejemplo, cuatro imágenes de referencia. Primero se comprueba una imagen de referencia coubicada y el resto de las imágenes candidatas se comprueban en el orden ascendente del índice de referencia (refldx). Cuando tanto $\text{RefPicList0}[\text{refldx}]$ como $\text{RefPicList1}[\text{refldx}]$ están disponibles, $\text{RefPicListX}[\text{refldx}]$ precede a la otra imagen, donde X es igual a $\text{collocated_from_I0_flag}$.

Para cada imagen candidata, se determinan tres regiones candidatas para derivar los bloques vecinos temporales. Cuando una región cubre más de un bloque de 16×16 , todos los bloques de 16×16 en dicha región se comprueban en orden de escaneo ráster. Las tres regiones candidatas se definen de la siguiente manera: CPU (la región coubicada del PU actual o el CU actual), CLCU (el conjunto de codificación más grande (LCU) que cubre la región coubicada del PU actual) y BR (el bloque 4×4 inferior derecho de la CPU).

El codificador de vídeo puede verificar los bloques vecinos espaciales y/o temporales en busca de un vector de disparidad en un orden particular. En algunos casos, el codificador de vídeo puede verificar primero los bloques vecinos espaciales (A_0 , A_1 , B_0 , B_1 y B_2), seguidos de los bloques vecinos temporales. Si uno de los bloques espacialmente vecinos tiene un vector de movimiento de disparidad, el codificador de vídeo puede terminar el procedimiento de comprobación y el codificador de vídeo puede usar el vector de movimiento de disparidad como el vector de disparidad final para el PU actual.

El codificador de vídeo puede comprobar cada una de las regiones candidatas de una imagen candidata. En un ejemplo, si la imagen candidata está en una primera vista sin base, el codificador de vídeo puede verificar las regiones candidatas en el orden de CPU, CLCU y BR. En este ejemplo, si la imagen candidata está en una segunda vista sin base, el codificador de vídeo puede comprobar las regiones candidatas en el orden de BR, CPU, CLCU.

En este ejemplo, la decodificación de imágenes asociadas a la primera vista sin base puede depender de la decodificación de imágenes asociadas a una vista base, pero no de imágenes asociadas a otras vistas. Además, en este ejemplo, la decodificación de imágenes asociadas con la segunda vista sin base también puede depender solo de la decodificación de imágenes asociadas con la vista base. En otros ejemplos, la decodificación de imágenes asociadas con la segunda vista sin base puede depender además de la primera vista sin base, pero no de las imágenes asociadas con otras vistas, si están presentes.

Cuando una región candidata cubre más de un bloque de 16×16 , el codificador de vídeo puede comprobar todos los bloques de 16×16 en la región candidata de acuerdo con un orden de exploración ráster. Cuando el codificador de vídeo comprueba una región candidata (o un bloque de 16×16 dentro de una región candidata), el codificador de vídeo puede determinar si un PU que cubre la región candidata especifica un vector de movimiento de disparidad. Si el PU que cubre la región candidata especifica un vector de movimiento de disparidad, el codificador de vídeo puede determinar el vector de disparidad del conjunto de vídeo actual basándose en el vector de movimiento de disparidad del PU.

La predicción de movimiento inter-vista se puede aplicar tanto a los modos AMVP como a los de fusión. Por ejemplo, como se señaló anteriormente, el modo AMVP se ha extendido de manera que se añade un predictor de vector de movimiento inter-vista a una lista de candidatos. En función del vector de disparidad derivado de NBDV, el codificador de vídeo determina un bloque de referencia en una vista de referencia añadiendo el vector de disparidad y la posición de la muestra media del bloque actual. Si el índice de referencia para el bloque actual se refiere a una imagen de referencia inter-vista, el codificador de vídeo puede establecer el predictor del vector de movimiento inter-vista igual al vector de disparidad correspondiente. Si el índice de referencia actual se refiere a una imagen de referencia temporal y el bloque de referencia utiliza una hipótesis de movimiento que se refiere a el mismo conjunto de acceso que el índice de referencia actual, el codificador de vídeo puede utilizar el vector de movimiento que está asociado con esta hipótesis de movimiento como un predictor de vector de movimiento inter-vista. En otros casos, el codificador de vídeo puede marcar el predictor de vector de movimiento inter-vista como no válido y el codificador de vídeo puede no incluir el vector de movimiento en la lista de candidatos a predictor de vector de movimiento.

Con respecto al modo de fusión/salto, la lista candidata de parámetros de movimiento se extiende mediante un conjunto de parámetros de movimiento que se obtiene usando predicción de movimiento inter-vista. Por ejemplo, el codificador de vídeo puede derivar un candidato de vector de movimiento del bloque de referencia en la vista de referencia de la misma manera que el modo AMVP indicado anteriormente. Si el vector de movimiento derivado es válido y su imagen de referencia tiene un valor de recuento de orden de imagen (POC) igual al de una entrada en la lista de imágenes de referencia del PU/CU actual, la información de movimiento (dirección de predicción, imágenes de referencia y vectores de movimiento) se puede añadir a la lista de candidatos de fusión después de convertir el índice de referencia en función del POC. Dicho candidato puede denominarse como un vector de movimiento predicho inter-vista. De lo contrario, el vector de disparidad se convierte en un vector de movimiento de disparidad inter-vista, que el codificador de vídeo puede agregar a la lista de candidatos de fusión, en la misma posición que un vector de movimiento predicho por inter-vista cuando esté disponible.

De manera similar a la predicción de movimiento inter-vista, la predicción residual inter-vista se basa en un vector de disparidad para cada CU, como se describe en mayor detalle con respecto a las Figuras 8 y 9 a continuación.

La Figura 8 es un diagrama conceptual que ilustra una estructura de predicción de ejemplo de codificación de vídeo multi-vista. Como ejemplo, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede codificar un bloque en vista V1 en el tiempo T_8 prediciendo el bloque usando el bloque P_e en vista V1 en el tiempo T_0 . El codificador de vídeo puede restar los valores de píxeles originales del bloque actual de P_e , obteniendo así las muestras residuales del bloque actual.

Además, el codificador de vídeo puede localizar un bloque de referencia en la vista de referencia (vista V0) mediante el vector de disparidad 104. Las diferencias de los valores de muestra originales del bloque de referencia I_b y sus muestras predichas P_b se denominan muestras residuales del bloque de referencia, como se indica por r_b en la ecuación a continuación. En algunos ejemplos, el codificador de vídeo puede restar r_b del residuo actual y solo transformar el código de la señal de diferencia resultante. Por lo tanto, cuando se utiliza la predicción residual de inter-vista, el bucle de compensación de movimiento puede expresarse mediante la siguiente ecuación:

$$\hat{I}_e = r_e + P_e + r_b$$

donde la reconstrucción del bloque actual \hat{I}_e es igual a los coeficientes descuantificados r_e más la predicción P_e y los coeficientes residuales normalizados de cuantificación r_b . Los codificadores de vídeo pueden tratar r_b como el predictor residual. Por lo tanto, de manera similar a la compensación de movimiento, r_b puede restarse del residuo actual y solo la señal de diferencia resultante se codifica por transformación.

El codificador de vídeo puede señalar condicionalmente un indicador para indicar el uso de la predicción residual inter-vista sobre una base de CU. Por ejemplo, el codificador de vídeo puede atravesar todas los conjuntos de transformación (TU) cubiertas o parcialmente cubiertas por la región de referencia residual. Si alguna de estas TU está intercodificada y contiene un valor de indicador de bloque codificado (CBF) distinto de cero (CBF de luma o CBF de croma), el codificador de vídeo puede marcar la referencia residual relacionada como disponible y el codificador de vídeo puede aplicar la predicción residual. En este caso, el codificador de vídeo puede señalar un indicador que indica el uso de la predicción residual inter-vista como parte de la sintaxis de CU. Si esta bandera es igual a 1, la señal residual actual se predice usando la señal residual de referencia potencialmente interpolada y solo la diferencia se transmite usando codificación de transformación. De lo contrario, el residuo del bloque actual se codifica convencionalmente utilizando la codificación de transformación HEVC.

La solicitud provisional de EE. UU. n.º 61/670.075, depositada el 10 de julio de 2012, y la solicitud provisional de EE. UU. n.º 61/706.692, depositada el 27 de septiembre de 2012, proponen una predicción residual generalizada (GRP) para la codificación de vídeo escalable. Aunque estas solicitudes de patente provisionales se centran en la codificación de vídeo escalable, las técnicas de GRP descritas en estas solicitudes de patente provisionales pueden ser aplicables a la codificación de vídeo multi-vista (por ejemplo, MV-HEVC y 3D-HEVC).

La idea general de GRP se puede formular, en el contexto de la unipredicción, como:

$$I_c = r_c + P_c + w * r_r$$

En la fórmula anterior, I_c denota la reconstrucción de un fotogramas actual en una capa (o vista) actual, P_c representa una predicción temporal de la misma capa (o vista), r_c indica un residuo señalado, r_r indica una predicción residual de una capa de referencia y w es un factor de ponderación. En algunos ejemplos, es posible que el factor de ponderación deba codificarse en un flujo de bits o derivarse en función de la información previamente codificada. Este marco para GRP se puede aplicar en casos de decodificación de un solo bucle y decodificación de múltiples bucles. La decodificación de bucle múltiple implica una versión no restringida de la predicción de un bloque utilizando la señal de menor resolución reconstruida y muestreada. Para decodificar un bloque en una capa de mejora, se debe acceder

a múltiples bloques en capas anteriores.

Por ejemplo, cuando el decodificador de vídeo 30 utiliza decodificación de múltiples bucles, el GRP se puede formular adicionalmente como:

$$I_c = r_c + P_c + w * (I_r - P_r),$$

En la fórmula anterior, P_r indica la predicción temporal para la imagen actual en la capa de referencia, P_c representa una predicción temporal de la misma capa (o vista), r_c indica un residuo señalizado, w es un factor de ponderación e I_r denota la reconstrucción completa de la imagen actual en la capa de referencia.

Las fórmulas anteriores incluyen un factor de ponderación que puede señalizarse en un flujo de bits o derivarse en función de la información codificada previamente. En algunos ejemplos, el codificador de vídeo 20 puede señalar, en un flujo de bits, en una base CU por CU, índices de ponderación utilizados en GRP. Cada índice de ponderación puede corresponder a un factor de ponderación que es mayor o igual que 0. Cuando un factor de ponderación para un CU actual es igual a 0, el bloque residual del CU actual se codifica utilizando la codificación de transformación HEVC convencional. De lo contrario, cuando el factor de ponderación para el CU actual es mayor que 0, la señal residual actual (es decir, el bloque residual para el CU actual) se puede predecir usando una señal residual de referencia multiplicada por el factor de ponderación y solo la diferencia se transmite usando codificación de transformación. En algunos ejemplos, la señal residual de referencia se interpola.

L. Zhang y col., "3D-CE5.h related: Advanced residual prediction for multiview coding," Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 2nd Meeting: Shanghai, CN, 13-19 de octubre de 2012, documento JCT3V-B0051 (en lo sucesivo, "JCT3V-B0051"), propuso un procedimiento avanzado de predicción residual (ARP) para mejorar aún más la eficiencia de codificación de la predicción residual inter- vista. En algunos casos, la ARP se puede realizar a un nivel de PU en lugar de un nivel de CU. Para distinguir el esquema de predicción residual descrito anteriormente de ARP, el esquema de predicción residual descrito anteriormente puede denominarse "predicción residual inter- vista basada en CU".

La Figura 9 es un diagrama conceptual que ilustra una estructura de predicción de ejemplo de ARP en codificación de vídeo multi- vista. La Figura 6 incluye cuatro imágenes: una imagen actual 110, una imagen de referencia temporal 112, una imagen de referencia de disparidad 114 y una imagen de referencia de disparidad temporal 116. La imagen actual 110 está asociada con la vista V1 y está asociada con la instancia de tiempo Tj. La imagen de referencia temporal 112 está asociada con la vista V1 y está asociada con la instancia de tiempo Ti. La imagen de referencia de disparidad 114 está asociada con la vista V0 y está asociada con la instancia de tiempo Tj. La imagen de referencia de disparidad temporal 116 está asociada con la vista V0 y está asociada con la instancia de tiempo Ti.

La imagen actual 110 incluye un PU actual indicada como "Dc." En otras palabras, Dc representa un bloque actual en una vista actual (vista 1). Dc tiene un vector de movimiento temporal V_D que indica una ubicación en la imagen de referencia temporal 112. El codificador de vídeo 20 puede determinar un bloque de referencia temporal D_r basándose en muestras en la imagen 112 que están asociadas con la ubicación indicada por el vector de movimiento temporal V_D. Por lo tanto, D_r denota el bloque de predicción temporal de Dc desde la misma vista (vista 1) en el tiempo Ti y V_D denota el movimiento de Dc a D_r.

Además, el codificador de vídeo 20 puede determinar un bloque de referencia de disparidad Bc en función de muestras en la imagen de referencia de disparidad 114 que están asociadas con una ubicación indicada por un vector de disparidad de Dc. Por lo tanto, Bc denota un bloque de referencia (es decir, la representación de Dc en la vista de referencia (vista 0) en el tiempo Tj). La posición superior izquierda de Bc se puede calcular con el vector de disparidad derivado añadiendo el vector de disparidad derivado a la posición superior izquierda de Dc. Dado que Dc y Bc pueden ser proyecciones del mismo objeto en dos vistas diferentes, Dc y Bc deben compartir la misma información de movimiento. Por lo tanto, el bloque de predicción temporal de Bc Br en la vista 0 en el tiempo Ti puede ubicarse desde Bc aplicando la información de movimiento de V_D.

El codificador de vídeo 20 puede determinar un bloque de referencia de disparidad temporal Br (el bloque predictivo de Bc) en la imagen de disparidad temporal 116. Como se indicó anteriormente, la imagen de disparidad temporal 116 está asociada con la misma vista (es decir, la vista V0) que Br y está asociada con la misma instancia de tiempo que D_r (es decir, la instancia de tiempo Ti). El codificador de vídeo 20 puede determinar Br en función de muestras en una ubicación indicada por el vector de movimiento V_D de Dc. Por lo tanto, la posición superior izquierda de Br se puede calcular con el vector de movimiento reutilizado V_D añadiendo el vector de movimiento V_D a la posición superior izquierda de Bc. La posición superior izquierda de Bc puede ser igual a la suma de la posición superior izquierda de Dc y el vector de disparidad. Por lo tanto, la posición superior izquierda de Br puede ser igual a la suma de las coordenadas de la posición superior izquierda de Dc, el vector de disparidad y el vector de movimiento V_D. De esta manera, como se muestra en la Figura 9 por la flecha 118, el codificador de vídeo 20 puede reutilizar el vector de movimiento V_D para determinar Br.

Además, en ARP, cada muestra en un primer bloque residual puede indicar la diferencia entre una muestra en D_c y una muestra correspondiente de D_r . El primer bloque residual puede denominarse como un bloque residual original para D_c . Cada muestra en un segundo bloque residual puede indicar una diferencia entre una muestra en B_c y una muestra correspondiente en B_r . El segundo bloque residual puede denominarse como un "predictor residual". Debido a que el codificador de vídeo 20 utiliza el vector de movimiento V_D para determinar B_r , el predictor residual puede ser diferente de los datos residuales reales de B_c .

Después de que el codificador de vídeo 20 determina el predictor residual, el codificador de vídeo 20 puede multiplicar el predictor residual por un factor de ponderación. En otras palabras, el residuo de B_c con información de movimiento de V_D se multiplica por un factor de ponderación y se utiliza como el predictor residual para el residuo actual. El factor de ponderación puede ser igual a 0, 0,5 o 1. Por lo tanto, se pueden usar tres factores de ponderación en ARP (es decir, 0, 0,5 y 1).

Después de que el codificador de vídeo 20 multiplica el predictor residual por el factor de ponderación, el predictor residual puede denominarse predictor residual ponderado. El codificador de vídeo 20 puede seleccionar, como factor de ponderación final, el factor de ponderación que conduce a un coste mínimo de tasa-distorsión para el CU actual (es decir, el CU que contiene el PU actual). El codificador de vídeo 20 puede incluir, en el flujo de bits, a nivel de CU, datos que indican un índice de ponderación. El índice de ponderación puede indicar el factor de ponderación final (es decir, el factor de ponderación que se utilizó para generar el predictor residual ponderado) para el CU actual. En algunos ejemplos, los índices de ponderación de 0, 1 y 2 corresponden a factores de ponderación de 0, 1 y 0,5, respectivamente. La selección del factor de ponderación de 0 para el CU actual es equivalente a no usar ARP para ninguna de los PU del CU actual.

El codificador de vídeo 20 puede entonces determinar un bloque residual final para el PU actual. Cada muestra en el bloque residual final para el PU actual puede indicar una diferencia entre una muestra en el bloque residual original y una muestra correspondiente en el predictor residual ponderado. Un bloque residual de un CU actual (es decir, el CU que contiene el PU actual) puede incluir el bloque residual final para el PU actual junto con bloques residuales, si los hay, para otras PU del CU actual. Como se describe en otra parte de esta divulgación, el codificador de vídeo 20 puede dividir el bloque residual del CU actual entre uno o más bloques de transformación. Cada uno de los bloques de transformación puede estar asociado con un TU del CU actual. Para cada bloque de transformación, el codificador de vídeo 20 puede aplicar una o más transformaciones al bloque de transformación para generar un bloque de coeficientes de transformación. El codificador de vídeo 20 puede incluir, en un flujo de bits, datos que representan coeficientes de transformación cuantificados del bloque de coeficientes de transformación.

Por lo tanto, en ARP, para garantizar una alta correlación entre los residuos de dos vistas, el codificador de vídeo 20 puede aplicar el movimiento de un PU actual a un bloque correspondiente en una imagen de vista de referencia para generar residuos en la vista base que se utilizará para la predicción residual inter-vista. De esta manera, el movimiento se alinea para el PU actual y el bloque de referencia correspondiente en la vista de referencia. Además, se aplica un factor de ponderación adaptativo a la señal residual para que el error de predicción se reduzca aún más.

Si el PU actual es bipredicha, el PU actual tiene un vector de movimiento $RefPicListO$, un vector de movimiento $RefPicList1$, un índice de referencia $RefPicListO$ y un índice de referencia $RefPicList1$. Esta descripción puede referirse a la imagen de referencia indicada por el índice de referencia $RefPicListO$ del PU actual como la imagen de referencia diana $RefPicListO$ del PU actual. El vector de movimiento $RefPicList1$ del PU actual puede indicar una ubicación de referencia en la imagen de referencia diana $RefPicList1$ del PU actual. Esta divulgación puede referirse a la imagen de referencia indicada por el índice de referencia $RefPicList1$ del PU actual como la imagen de referencia diana $RefPicList1$ del PU actual. El vector de movimiento $RefPicList1$ del PU actual puede indicar una ubicación de referencia en la imagen de referencia diana $RefPicList1$ del PU actual.

Por lo tanto, cuando el codificador de vídeo 20 realiza ARP en un PU bipredicha, el codificador de vídeo 20 puede determinar, basándose en el vector de movimiento $RefPicListO$ del PU actual, una ubicación de referencia en la imagen de referencia diana $RefPicListO$ del PU actual. Esta divulgación puede referirse a esta ubicación de referencia como la ubicación de referencia $RefPicListO$ del PU actual. El codificador de vídeo 20 puede entonces determinar un bloque de referencia que incluye muestras reales o interpoladas de la imagen de referencia diana $RefPicListO$ del PU actual que están asociadas con la ubicación de referencia $RefPicListO$ del PU actual. Esta divulgación puede referirse a este bloque de referencia como el bloque de referencia $RefPicListO$ del PU actual.

Además, el codificador de vídeo 20 puede determinar, en función del vector de movimiento $RefPicList1$ del PU actual, una ubicación de referencia en la imagen de referencia diana $RefPicList1$ del PU actual. Esta divulgación puede referirse a esta ubicación de referencia como la ubicación de referencia $RefPicList1$ del PU actual. El codificador de vídeo 20 puede entonces determinar un bloque de referencia que incluye muestras reales o interpoladas de la imagen de referencia diana $RefPicList1$ del PU actual que están asociadas con la ubicación de referencia $RefPicListO$ del PU actual. Esta divulgación puede referirse a este bloque de referencia como el bloque de referencia $RefPicList1$ del PU actual.

El codificador de vídeo 20 puede determinar, en función del bloque de referencia RefPicListO del PU actual y el bloque de referencia RefPicList1 del PU actual, un bloque predictivo temporal para el PU actual. Por ejemplo, cada muestra en el bloque predictivo temporal del PU actual puede indicar un promedio ponderado de las muestras correspondientes en el bloque de referencia RefPicListO del PU actual y el bloque de referencia RefPicList1 del PU actual.

Además, cuando el codificador de vídeo 20 realiza ARP en un PU bipredicha, el codificador de vídeo 20 puede determinar, en función del vector de movimiento RefPicListO del PU actual y una ubicación dentro de un fotograma de referencia de disparidad de un bloque de referencia de disparidad, una ubicación de referencia de disparidad temporal en una imagen de referencia de disparidad temporal. Esta divulgación puede referirse a esta ubicación de referencia de disparidad temporal y esta imagen de referencia de disparidad temporal como la ubicación de referencia de disparidad temporal RefPicListO y la imagen de referencia de disparidad temporal RefPicListO, respectivamente. La imagen de referencia de disparidad temporal RefPicListO puede tener el mismo valor de POC que la imagen de referencia diana RefPicListO del PU actual. El codificador de vídeo 20 puede entonces determinar un bloque de muestra que incluye muestras reales o interpoladas de la imagen de referencia de disparidad temporal RefPicListO que están asociadas con la ubicación de referencia de disparidad temporal RefPicListO. Esta divulgación puede referirse a este bloque de muestra como el bloque de referencia de disparidad temporal RefPicListO.

Además, el codificador de vídeo 20 puede determinar, en función del vector de movimiento RefPicList1 del PU actual y la ubicación dentro del fotograma de referencia de disparidad del bloque de referencia de disparidad, una ubicación de referencia de disparidad temporal en una imagen de referencia de disparidad temporal. Esta divulgación puede referirse a esta ubicación de referencia de disparidad temporal y esta imagen de referencia de disparidad temporal como la ubicación de referencia de disparidad temporal RefPicList1 y la imagen de referencia de disparidad temporal RefPicList1, respectivamente. La imagen de referencia de disparidad temporal RefPicList1 puede tener el mismo valor de POC que la imagen de referencia diana RefPicList1 del PU actual. Debido a que la imagen de referencia diana RefPicListO del PU actual y la imagen de referencia diana RefPicList1 del PU actual pueden ser diferentes, la imagen de referencia de disparidad temporal RefPicList1 puede ser diferente de la imagen de referencia de disparidad temporal RefPicListO. El codificador de vídeo 20 puede entonces determinar un bloque de muestra que incluye muestras reales o interpoladas de la imagen de referencia de disparidad temporal RefPicList1 que están asociadas con la ubicación de referencia de disparidad temporal RefPicList1. Esta divulgación puede referirse a este bloque de muestra como el bloque de referencia de disparidad temporal RefPicList1.

A continuación, el codificador de vídeo 20 puede determinar, en función del bloque de referencia de disparidad temporal RefPicListO y el bloque de referencia de disparidad temporal RefPicList1, un bloque predictivo de disparidad. En algunos ejemplos, cada muestra en el bloque predictivo de disparidad es un promedio ponderado de las muestras correspondientes en el bloque de referencia de disparidad temporal RefPicListO y el bloque de referencia de disparidad temporal RefPicList1. El codificador de vídeo 20 puede entonces determinar un predictor residual. El predictor residual puede ser un bloque de muestras. Cada muestra en el predictor residual puede indicar una diferencia entre una muestra en el bloque de referencia de disparidad y una muestra correspondiente en el bloque predictivo de disparidad. El codificador de vídeo 20 puede entonces generar un predictor residual ponderado mediante la aplicación de un factor de ponderación al predictor residual. El codificador de vídeo 20 puede entonces determinar un bloque residual final para el PU actual. Cada muestra en el bloque residual final del PU actual puede indicar una diferencia entre una muestra en el bloque de predicción original para el PU actual y las muestras correspondientes en el bloque predictivo temporal del PU actual y el predictor residual ponderado. El codificador de vídeo 20 puede señalar el bloque residual final del PU actual en el flujo de bits.

El decodificador de vídeo 30 puede realizar un procedimiento similar cuando realiza ARP en un PU y un PU bipredicha. Por ejemplo, el decodificador de vídeo 30 puede determinar el bloque predictivo temporal del PU actual y el predictor residual ponderado de la manera de muestra descrita anteriormente. El decodificador de vídeo 30 puede determinar el bloque residual final del PU actual en función de los datos señalizados en el flujo de bits. El decodificador de vídeo 30 puede entonces reconstruir el bloque de predicción del PU actual añadiendo el bloque residual final del PU actual, el bloque predictivo temporal del PU actual y el predictor residual ponderado.

La Figura 10 ilustra la relación entre el bloque actual, el bloque correspondiente y el bloque compensado por movimiento descrito anteriormente. En otras palabras, la Figura 10 es un diagrama conceptual que ilustra una relación de ejemplo entre un bloque actual, un bloque de referencia y un bloque compensado por movimiento en ARP. En el ejemplo de la Figura 10, un codificador de vídeo está codificando actualmente un PU 131 actual en una imagen 131 actual. La imagen actual 131 está asociada con una vista V1 y una instancia de tiempo T1.

Además, en el ejemplo de la Figura 10, el codificador de vídeo puede determinar un bloque de referencia 132 (es decir, un bloque correspondiente) que comprende muestras reales o interpoladas de una imagen de referencia 133 que están asociadas con una ubicación indicada por un vector de disparidad del PU actual 130. Por ejemplo, una esquina superior izquierda del bloque de referencia 132 puede ser la ubicación indicada por el vector de disparidad del PU actual 130. El bloque de referencia de disparidad temporal 145 puede tener el mismo tamaño que el bloque de predicción del PU actual 130.

En el ejemplo de la Figura 10, el PU actual 130 tiene un primer vector de movimiento 134 y un segundo vector de movimiento 136. El vector de movimiento 134 indica una ubicación en la imagen de referencia temporal 138. La imagen de referencia temporal 138 está asociada con la vista V1 (es decir, la misma vista que la imagen actual 131) y una instancia de tiempo T0. El vector de movimiento 136 indica una ubicación en la imagen de referencia temporal 140. La imagen de referencia temporal 140 está asociada con la vista V1 y una instancia de tiempo T3.

De acuerdo con el esquema ARP descrito anteriormente, el codificador de vídeo puede determinar una imagen de referencia (es decir, la imagen de referencia 142) que está asociada con la misma vista que la imagen de referencia 133 y está asociada con la misma instancia de tiempo que la imagen de referencia temporal 138. Además, el codificador de vídeo puede añadir el vector de movimiento 134 a las coordenadas de una esquina superior izquierda del bloque de referencia 132 para derivar una ubicación de referencia de disparidad temporal. El codificador de vídeo puede determinar un bloque de referencia de disparidad temporal 143 (es decir, un bloque compensado por movimiento). Las muestras en el bloque de referencia de disparidad temporal 143 pueden ser muestras reales o interpoladas de la imagen de referencia 142 que están asociadas con la ubicación de referencia de disparidad temporal derivada del vector de movimiento 134. El bloque de referencia de disparidad temporal 143 puede tener el mismo tamaño que el bloque de predicción del PU actual 130.

De manera similar, el codificador de vídeo puede determinar una imagen de referencia (es decir, la imagen de referencia 144) que está asociada con la misma vista que la imagen de referencia 134 y está asociada con la misma instancia de tiempo que la imagen de referencia temporal 140. Además, el codificador de vídeo puede agregar el vector de movimiento 136 a las coordenadas de la esquina superior izquierda del bloque de referencia 132 para derivar una ubicación de referencia de disparidad temporal. El codificador de vídeo puede entonces determinar un bloque de referencia de disparidad temporal 145 (es decir, un bloque compensado por movimiento). Las muestras en el bloque de referencia de disparidad temporal 145 pueden ser muestras reales o interpoladas de la imagen de referencia 144 que están asociadas con la ubicación de referencia de disparidad temporal derivada del vector de movimiento 136. El bloque de referencia de disparidad temporal 145 puede tener el mismo tamaño que el bloque de predicción del PU actual 130.

Además, en el ejemplo de la Figura 10, el codificador de vídeo puede determinar, basándose en el bloque de referencia de disparidad temporal 143 y el bloque de referencia de disparidad temporal 145, un bloque predictivo de disparidad. El codificador de vídeo puede entonces determinar un predictor residual. Cada muestra en el predictor residual puede indicar una diferencia entre una muestra en el bloque de referencia 132 y una muestra correspondiente en el bloque predictivo de disparidad.

De acuerdo con los aspectos de esta divulgación, un codificador de vídeo (tal como un codificador de vídeo o un decodificador de vídeo) puede habilitar o deshabilitar ARP (incluida la codificación residual de una capa en relación con la residual de una segunda capa diferente) en función de las imágenes de referencia en las listas de imágenes de referencia para el bloque que se está codificando actualmente. En un ejemplo, el codificador de vídeo puede habilitar o deshabilitar ARP en función de si las listas de imágenes de referencia para el bloque que se está codificando actualmente incluyen imágenes de referencia temporales. De acuerdo con los aspectos de esta descripción, si las listas de imágenes de referencia para un bloque inter-predicho incluyen solo imágenes de referencia inter-vista, el codificador de vídeo puede deshabilitar ARP. En dicho ejemplo, cuando el codificador de vídeo comprende un codificador de vídeo, el codificador de vídeo puede no señalar un factor de ponderación en el flujo de bits (omitir la señalización de un factor de ponderación). Del mismo modo, cuando el codificador de vídeo comprende un decodificador de vídeo, el decodificador de vídeo también puede omitir la decodificación de un factor de ponderación e inferir que el factor de ponderación es igual a cero.

Las técnicas descritas anteriormente se pueden aplicar en el contexto de imágenes de acceso aleatorio. Por ejemplo, de acuerdo con los aspectos de esta descripción, el codificador de vídeo puede habilitar o deshabilitar ARP en función de si el componente de vista que se está codificando actualmente es un componente de vista de acceso aleatorio.

Con respecto a los componentes de vista de acceso aleatorio, en HEVC, en general, hay cuatro tipos de imágenes que se pueden identificar por el tipo de conjunto NAL. Los cuatro tipos de imágenes incluyen una imagen de actualización de decodificación instantánea (IDR), una imagen CRA, una imagen de acceso de capa temporal (TLA) y una imagen codificada que no es una imagen IDR, CRA o TLA. La IDR y las imágenes codificadas son tipos de imágenes heredados de la especificación H.264/AVC. Los tipos de imagen CRA y TLA son nuevas incorporaciones para el estándar HEVC. Una imagen CRA es un tipo de imagen que facilita la decodificación que comienza desde cualquier punto de acceso aleatorio en el centro de una secuencia de vídeo, y puede ser más eficiente que insertar imágenes IDR. Una imagen TLA es un tipo de imagen que se puede usar para indicar puntos de cambio de capa temporal válidos.

En aplicaciones de vídeo, como la radiodifusión y la transmisión (*streaming*), la conmutación puede ocurrir entre diferentes canales de datos de vídeo y el salto puede ocurrir a partes específicas de los datos de vídeo. En tales casos, puede ser beneficioso lograr un retraso mínimo durante el cambio y/o el salto. Esta función se habilita al tener imágenes

de acceso aleatorio a intervalos regulares en los flujos de bits de vídeo. La imagen IDR, especificada tanto en H.264/AVC como en HEVC, se puede usar para acceso aleatorio. Sin embargo, una imagen IDR inicia una secuencia de vídeo codificada y elimina imágenes de un búfer de imágenes decodificadas (DPB) (que también puede denominarse memoria de imágenes de referencia, como se describe a continuación con respecto a las Figuras 2 y 3).

Por consiguiente, las imágenes que siguen a la imagen IDR en orden de decodificación no pueden usar imágenes decodificadas antes de la imagen IDR como referencia. En consecuencia, los flujos de bits que dependen de imágenes IDR para acceso aleatorio pueden tener una menor eficiencia de codificación. Para mejorar la eficiencia de codificación, las imágenes CRA en HEVC permiten que las imágenes que siguen a una imagen CRA en orden de decodificación pero que preceden a la imagen CRA en orden de salida utilicen imágenes decodificadas antes de la imagen CRA como referencia.

En HEVC, un flujo de bits que comienza con una imagen CRA se considera un flujo de bits conforme. Cuando un flujo de bits comienza con una imagen CRA, las imágenes principales de la imagen CRA pueden referirse a imágenes de referencia no disponibles y, por lo tanto, pueden no decodificarse correctamente. Sin embargo, HEVC especifica que las imágenes principales de la imagen CRA inicial no se emiten, de ahí el nombre "acceso aleatorio limpio". Para el establecimiento del requisito de conformidad del flujo de bits, HEVC especifica un procedimiento de decodificación para generar imágenes de referencia no disponibles para la decodificación de las imágenes principales sin salida. Sin embargo, las implementaciones de decodificador conformes no tienen que seguir ese procedimiento de decodificación, siempre que estos decodificadores conformes puedan generar una salida idéntica en comparación con cuando el procedimiento de decodificación se realiza desde el comienzo del flujo de bits. En HEVC, un flujo de bits conforme puede no contener imágenes IDR en absoluto y, en consecuencia, puede contener un subconjunto de una secuencia de vídeo codificada o una secuencia de vídeo codificada incompleta.

Además de las imágenes IDR y CRA, hay otros tipos de imágenes de puntos de acceso aleatorio, por ejemplo, una imagen de acceso de enlace roto (BLA). Para cada uno de los tipos principales de las imágenes de punto de acceso aleatorio, puede haber subtipos, dependiendo de cómo los sistemas podrían tratar potencialmente una imagen de punto de acceso aleatorio. Cada subtipo de imagen de punto de acceso aleatorio tiene un tipo de conjunto NAL diferente.

En general, con respecto a las extensiones de HEVC (como MV-HEVC, 3D-HEVC o SHVC), si un componente de vista es un punto de acceso aleatorio puede depender del tipo de conjunto NAL del componente de vista. Si el tipo pertenece a los definidos en la especificación base de HEVC para imágenes de punto de acceso aleatorio, el componente de vista actual es un componente de vista de punto de acceso aleatorio (o, por simplicidad, imagen de punto de acceso aleatorio de la vista actual).

En algunos casos, la funcionalidad de acceso aleatorio solo se aplica a la predicción temporal de una manera que ciertas predicciones en la dimensión temporal (por lo tanto, dentro de una vista) están deshabilitadas o restringidas de manera similar a la especificación base de HEVC. Sin embargo, la predicción inter-vista para un componente de vista de punto de acceso aleatorio todavía es posible, y generalmente se realiza para mejorar la eficiencia de codificación, similar a la imagen de anclaje en H.264/MVC. Por lo tanto, un componente de vista de punto de acceso aleatorio (RAP), si se utiliza la predicción inter-vista, puede ser una imagen P o B.

De acuerdo con los aspectos de esta divulgación, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede deshabilitar la predicción residual inter-vista para cada bloque de un componente de vista de acceso aleatorio. En dichos ejemplos, el codificador de vídeo 20 puede no señalar un factor de ponderación en el flujo de bits (omitir la señalización de un factor de ponderación). El decodificador de vídeo 30 también puede omitir la decodificación de un factor de ponderación y determinar automáticamente que el factor de ponderación es igual a cero.

En otro ejemplo, de acuerdo con aspectos de esta divulgación, el codificador de vídeo puede habilitar ARP si al menos una imagen de referencia es de la misma vista que el bloque que se está codificando actualmente. Adicional o alternativamente, el codificador de vídeo puede habilitar ARP solo cuando ambas imágenes de referencia (correspondientes a una imagen de referencia en RefPicList0 y una imagen de referencia en RefPicList1), si están disponibles, son de la misma vista que el bloque que se está codificando actualmente. Adicional o alternativamente, el codificador de vídeo puede deshabilitar ARP para un bloque si el bloque está codificado inter-vista con una imagen de referencia inter-vista. Como se señaló anteriormente, cuando ARP está desactivado, el factor de ponderación no se señala.

En algunos ejemplos, cuando un búfer de imágenes decodificadas para codificar un bloque actual no incluye una imagen en la misma vista que la imagen de referencia de disparidad que tiene el mismo POC que la imagen de referencia temporal, el codificador de vídeo puede modificar el procedimiento ARP.

En otro ejemplo, adicional o alternativamente, cuando una o ambas de las listas de imágenes de referencia del bloque de referencia de disparidad no incluyen una imagen de referencia en la misma vista que la imagen de referencia de disparidad que tiene el mismo POC que la imagen de referencia temporal, el codificador de vídeo puede modificar el

procedimiento ARP. Por ejemplo, dado un índice de lista de imágenes de referencia actual X (siendo X 0 o 1) para el fragmento que contiene el bloque de referencia de disparidad, en un ejemplo, si la lista de imágenes de referencia con un índice de lista igual a X del bloque de referencia de disparidad no incluye una imagen de referencia en la misma vista que la imagen de referencia de disparidad y que tiene el mismo POC que la imagen de referencia temporal del bloque actual, el codificador de vídeo puede modificar el procedimiento ARP. En otro ejemplo, si ninguna de las listas de imágenes de referencia (por ejemplo, ni la lista 0 ni la lista 1) del bloque de referencia de disparidad incluye una imagen de referencia en la misma vista que la imagen de referencia de disparidad y que tiene el mismo POC que la imagen de referencia temporal del bloque actual, el codificador de vídeo puede modificar el procedimiento ARP.

En algunos ejemplos, el codificador de vídeo puede modificar el procedimiento ARP deshabilitando el procedimiento ARP, de modo que el bloque actual no esté codificado usando ARP. En otros ejemplos, el codificador de vídeo puede modificar el procedimiento ARP escalando el vector de movimiento temporal para identificar otra imagen de referencia de disparidad temporal. Por ejemplo, el codificador de vídeo puede escalar el vector de movimiento temporal de manera que, cuando se combina con el vector de disparidad, la combinación escalada es una imagen de referencia de temporaldisparidad que se incluye en la lista de imágenes de referencia y está en una ubicación temporalmente más cercana a la imagen de referencia de disparidad. Las técnicas descritas anteriormente pueden evitar que el codificador de vídeo intente ubicar el bloque de referencia de disparidad en una imagen que no está incluida en el búfer de imágenes decodificadas o una o ambas listas de imágenes de referencia.

De acuerdo con otros aspectos de esta descripción, ARP puede habilitarse o deshabilitarse en función de un modo de partición y/o modo de codificación del bloque que se está codificando actualmente. Por ejemplo, los factores de ponderación solo pueden señalizarse para ciertos modos de partición y/o ciertos modos de codificación. Si un factor de ponderación no se incluye en un flujo de bits, un decodificador de vídeo puede omitir la decodificación del factor de ponderación e inferir que el factor de ponderación es de valor cero (desactivando así ARP). De acuerdo con aspectos de esta descripción, en algunos ejemplos, el factor de ponderación para cualquier bloque intercodificado con modo de partición desigual a PART_2Nx2N puede no señalizarse. En otro ejemplo, puede no señalizarse el factor de ponderación para un bloque intercodificado con un modo de partición distinto de PART_2Nx2N, PART_2NxN y PART_Nx2N. En incluso otro ejemplo, adicional o alternativamente, el factor de ponderación para cualquier bloque intercodificado con modo de codificación desigual al modo de salto y/o modo de fusión puede no señalizarse.

De acuerdo con aún otros aspectos de esta descripción, se puede implementar un enfoque más flexible para los factores de ponderación. Por ejemplo, la cantidad de factores de ponderación disponibles puede alterarse a nivel de secuencia (por ejemplo, en un conjunto de parámetros, tal como un conjunto de parámetros de secuencia (SPS)). En un ejemplo con fines ilustrativos, un indicador puede señalizarse en un SPS para desactivar uno o más factores de ponderación, por ejemplo, 0,5 y/o 1. En otro ejemplo, dicho indicador puede señalizarse en VPS y aplicarse a todas las vistas sin base. En otro ejemplo más, dicho indicador puede señalizarse en la extensión del conjunto de parámetros de vídeo (VPS) para cada vista sin base. En otro ejemplo, dicho indicador puede proporcionarse en un conjunto de parámetros de imagen (PPS), un encabezado de fragmento o un conjunto de parámetros de vista para deshabilitar uno o más factores de ponderación. Cuando se ha deshabilitado un factor de ponderación, se pueden usar menos bits para representar los factores de ponderación restantes, proporcionando así un ahorro de bits.

Según otros aspectos, se puede proporcionar un indicador para modificar y/o reemplazar uno o más factores de ponderación. En un ejemplo, el codificador de vídeo puede reemplazar el factor de ponderación de 0,5 por un factor de ponderación de 0,75. Este indicador puede señalizarse en un encabezado de fragmento, un SPS, un conjunto de parámetros de imagen (PPS) o un VPS.

De acuerdo con los aspectos de esta divulgación, en una implementación de ejemplo, el codificador de vídeo puede usar un procedimiento de predicción residual inter-vista modificado como el descrito en 3D-HTM versión 5.0 (indicado anteriormente). Por ejemplo, de acuerdo con los aspectos de esta divulgación, se pueden usar uno o más elementos sintácticos para indicar que se aplica la predicción residual inter-vista.

En un ejemplo, uno o más elementos sintácticos que indican un índice de un factor de ponderación (por ejemplo, un elemento sintáctico `weighting_factor_index`) pueden señalizarse como parte del CU. En este ejemplo, la sintaxis del CU puede modificarse (por ejemplo, en relación con la versión 5.0 de 3D-HTM) y el elemento de sintaxis del factor de ponderación puede señalizarse solo cuando se cumplen las siguientes condiciones: la vista actual es una vista de textura dependiente, el CU actual no está intracodificada y el CU actual tiene un modo de partición igual a PART_2Nx2N. Cuando este elemento de sintaxis no está presente en el flujo de bits, se infiere que el factor de ponderación es igual a 0. Una tabla de sintaxis de CU de ejemplo se muestra a continuación:

coding_unit(x0, y0, log2CbSize , ctDepth) {	Descriptor
if(transquant_bypass_enable_flag) {	
cu_transquant_bypass_flag	ae(v)
}	
...	
} else { /* PART_NxN */	
prediction_unit(x0, y0 , log2CUSize)	
prediction_unit(x1, y0 , log2CUSize)	
prediction_unit(x0, y1 , log2CUSize)	
prediction_unit(x1, y1 , log2CUSize)	
}	
if (!depth_flag && layer_id && PredMode != MODE_INTRA && PartMode == PART_2Nx2N)	
weighting_factor_index	ae(v)
...	
if(!pcm_flag) {	
if(PredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]))	
no_residual_syntax_flag	ae(v)
if(!no_residual_syntax_flag) {	
MaxTrafoDepth = (PredMode[x0][y0] == MODE_INTRA ? max_transform_hierarchy_depth_intra + IntraSplitFlag : max_transform_hierarchy_depth_inter)	
transform_tree(x0, y0 x0, y0, log2CbSize, 0, 0)	
}	
}	
}	
}	

Otra tabla de sintaxis de CU de ejemplo se muestra a continuación:

coding_unit(x0, y0, log2CbSize , ctDepth) {	Descriptor
if(transquant_bypass_enable_flag) {	
cu_transquant_bypass_flag	ae(v)
}	
} else { /* PART_NxN */	
prediction_unit(x0, y0 , log2CUSize)	
prediction_unit(x1, y0 , log2CUSize)	
prediction_unit(x0, y1 , log2CUSize)	
prediction_unit(x1, y1 , log2CUSize)	

(continuación)

	Descriptor
<code>coding_unit(x0, y0, log2CbSize , ctDepth) {</code>	
<code>}</code>	
<code>if (!depth_flag && layer_id && PredMode != MODE_INTRA && PartMode == PART_2Nx2N && TempMVAvai && DispVectAvai)</code>	
<i>weighting_factor_index</i>	<i>ae(v)</i>
<code>...</code>	
<code>if(!pcm_flag) {</code>	
<code>if(PredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]))</code>	
<i>no_residual_syntax_flag</i>	<i>ae(v)</i>
<code>if(!no_residual_syntax_flag) {</code>	
<code>MaxTrafoDepth = (PredMode[x0][y0] == MODE_INTRA ?</code> <code>max_transform_hierarchy_depth_intra + IntraSplitFlag :</code> <code>max_transform_hierarchy_depth_inter)</code>	
<code>transform_tree(x0, y0 x0, y0, log2CbSize, 0, 0)</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	

En el ejemplo anterior, TempMVAvai se puede establecer igual a 1 cuando el CU actual se predice a partir de al menos una imagen de referencia de la misma vista. De lo contrario, se establece igual a 0. Además, DispVectAvai puede establecerse igual a 1 si se puede encontrar un vector de disparidad. De lo contrario, es igual a 0.

En otro ejemplo, el elemento de sintaxis de factor de ponderación puede señalizarse solo cuando se cumplen las siguientes condiciones: la vista actual es una vista de textura dependiente, el CU actual no está intracodificado y el CU actual tiene un modo de partición igual a PART_2Nx2N, el vector de disparidad derivado está disponible y al menos una partición tiene un vector de movimiento temporal, por ejemplo, la imagen de referencia es de la misma vista. Cuando este elemento de sintaxis no está presente en el flujo de bits, se infiere que el factor de ponderación es igual a 0.

En otro ejemplo más, el elemento de sintaxis de factor de ponderación puede señalizarse solo cuando se cumplen las siguientes condiciones: la vista actual es una vista de textura dependiente, el CU actual no está intracodificado y el CU actual tiene un modo de partición igual a PART_2Nx2N, el vector de disparidad derivado está disponible y al menos una partición en todos los PU del CU actual tiene un vector de movimiento temporal, por ejemplo, la imagen de referencia es de la misma vista. Cuando este elemento de sintaxis no está presente en el flujo de bits, se infiere que el factor de ponderación es igual a 0.

En otro ejemplo más, el elemento de sintaxis de factor de ponderación puede señalizarse solo cuando se cumplen las siguientes condiciones: la vista actual es una vista de textura dependiente y el vector de disparidad derivado está disponible.

De acuerdo con los aspectos de la presente divulgación, el factor de ponderación puede señalizarse de diversas maneras. Por ejemplo, como se señaló anteriormente, el elemento de sintaxis *weighting_factor_index* puede indicar un índice para un factor de ponderación utilizado para la predicción residual avanzada. Cuando no está presente, la predicción residual avanzada puede deshabilitarse para el CU actual. Por ejemplo, si el factor de ponderación es igual a 0, el residuo del bloque actual se codifica convencionalmente utilizando la codificación de transformación HEVC, y las especificaciones, como en la subcláusula 8.5.2.2, de la especificación HEVC (por ejemplo, como WD9, como se identificó anteriormente) se invoca para obtener las muestras de predicción. Si el índice del factor de ponderación está presente, la señal residual actual se predice utilizando la señal residual de referencia potencialmente interpolada multiplicada por el factor de ponderación y solo se transmite la diferencia, y el procedimiento descrito a continuación con respecto a las subcláusulas modificadas 8.5.2.2.1 y 8.5.2.2.2 de la especificación HEVC (por ejemplo, WD9) se puede invocar para cada lista de predicción donde se utiliza la imagen de referencia temporal.

En algunos casos, un índice de factor de ponderación puede asignarse a un factor de ponderación. De esta manera, el codificador de vídeo puede implementar un enfoque más flexible para ponderar los factores en la predicción residual inter-vista. Por ejemplo, suponga con fines ilustrativos que hay N factores de ponderación diferentes que deben señalizarse, con N igual a 2, 3, 4 o similares. Cada uno de estos factores de ponderación se puede asignar inicialmente a un índice de ponderación único, como se muestra en el ejemplo de la Tabla 1 a continuación, donde $W_0, W_1, W_2, \dots, W_{N-1}$ son los factores de ponderación en el orden ascendente de los valores.

Tabla 1 - Mapeo entre índices de factores de ponderación y factores de ponderación

Valor de weighting_factor_index	Valor de los factores de ponderación
0	W_g
1	W_1
2	W_2
3	...
4	...
5	...
...	...
$N-1$	W_{N-1}

En otro ejemplo, $W_0, W_1, W_2, \dots, W_{N-1}$ pueden representar los factores de ponderación en orden descendente de la probabilidad de que se utilice el factor de ponderación, que se puede calcular durante la codificación.

Otro mapeo de ejemplo se muestra en la Tabla 2 a continuación, donde los factores de ponderación iguales a 0, 1, 0,5 están indexados por 0, 1, 2, respectivamente. Todos los factores de ponderación restantes pueden indexarse en función del orden ascendente de los valores o del orden descendente de las probabilidades.

Tabla 2 - Mapeo entre índices de factores de ponderación y factores de ponderación

Valor de weighting_factor_index	Valor de los factores de ponderación
0	0
1	1
2	0,5
3	...
4	...
5	...
...	...
$N-1$...

El decodificador de vídeo puede analizar el índice de factor de ponderación de un flujo de bits codificado para determinar el valor del índice. En un ejemplo, cada factor de ponderación puede identificarse mediante un índice de factor de ponderación, y los índices de factor de ponderación pueden señalizarse utilizando binarización unaria truncada, como se describe en la sección 9.3.2.2 de la especificación HEVC (por ejemplo, WD9). En otro ejemplo, los factores de ponderación pueden mapearse primero a un índice de ponderación único en función del orden descendente de las probabilidades de los factores de ponderación y luego codificarse con binarización unaria truncada.

En otro ejemplo más, el procedimiento de binarización puede definirse de acuerdo con la Tabla 3 a continuación:

Tabla 3 - Binarización de factores de ponderación (N es mayor que 4)

Valor de Weighting_factor_index	Valores bin.						
0	0						
1	1	0	0				
2	1	0	1				
3	1	1	0				
4	1	1	1	0			
5	1	1	1	1	0		
	1	1	1	1	1	0	
N-1	1	1	1	1	1	...	1
binIdx	0	1	2	3	4	...	N-3

Aquí, la cadena binaria de índices de factores de ponderación correspondientes a los valores 3 a N-1 consiste en un prefijo de '11' y un sufijo, indexados restando 3 del valor de weighting_factor_index donde se utiliza la binarización unaria truncada.

Cuando hay un total de cuatro factores de ponderación, el procedimiento de binarización se puede definir mediante la siguiente tabla:

Tabla 4 - Binarización de factores de ponderación (N = 4)

Valor de Weighting_factor_index	Valores bin.		
0	0		
1	1	0	0
2	1	0	1
3	1	1	
binIdx	0	1	2

Cuando hay en total tres factores de ponderación, por ejemplo, 0, 0,5 y 1, el procedimiento de binarización se puede definir mediante la siguiente tabla:

Tabla 5 - Binarización de factores de ponderación (N = 3)

Valor de Weighting_factor_index	Contenedor bin.	
0	0	
1	1	0
2	1	1
binIdx	0	1

Con respecto a la inicialización de contexto, se puede usar un conjunto de contextos para codificar los índices de factor de ponderación. El modo avanzado de predicción residual inter-venta puede aplicarse tanto a los fragmentos P como a los fragmentos B. La probabilidad inicial para el contexto de ponderación de los índices de los fragmentos P puede ser diferente de la de los fragmentos B. Alternativamente, todos los modelos de contexto se inicializan con igual probabilidad para diferentes valores bin, por ejemplo, 0 y 1.

Con respecto a la selección de contexto, suponga con fines ilustrativos que la ubicación de luma (xC, yC) especifica la muestra de luma superior izquierda del bloque de codificación de luma actual en relación con la muestra superior izquierda de la imagen actual. Supongamos además que la variable availableL, que especifica la disponibilidad del bloque de codificación ubicado directamente a la izquierda del bloque de codificación actual, se deriva invocando el procedimiento de derivación de disponibilidad para un bloque en orden de escaneo z como se especifica en la subcláusula 6.4.1 de la especificación HEVC con la ubicación (xCurr, yCurr) establecida igual a (xC, yC) y la ubicación vecina (xN, yN) establecida igual a (xC - 1, yC) como la entrada y la salida se asigna a availableL.

En el ejemplo anterior, la variable `availableA` que especifica la disponibilidad del bloque de codificación ubicado directamente encima del bloque de codificación actual, se puede derivar invocando el procedimiento de derivación de disponibilidad para un bloque en orden de escaneo *z* como se especifica en la subcláusula 6.4.1 de la especificación HEVC (por ejemplo, WD9) con la ubicación (`xCurr`, `yCurr`) establecida igual a (`xC`, `yC`) y la ubicación vecina (`xN`, `yN`) establecida igual a (`xC`, `yC - 1`) como la entrada y la salida se asigna a `availableA`.

De acuerdo con los aspectos de esta descripción, `condTermFlagN` (*N* puede ser *L* o *A*) se puede derivar de la siguiente manera:

- Si `mbPAddrN` está disponible y el factor de ponderación para el bloque `mbPAddrN` es desigual a 0, `condTermFlagN` se establece igual a 1
- De lo contrario (`mbPAddrN` no está disponible o el factor de ponderación para el bloque `mbPAddrN` es igual a 0), `condTermFlagN` se establece igual a 0.

Además, suponga que el `ctxIdx` es el índice de contexto que se utilizará para codificar los índices de factores de ponderación. En este ejemplo, los incrementos `ctxIdx` (`ctxIdxInc`) para cada bin a codificar se derivan mediante $\text{ctxIdxInc} = M * \text{condTermFlagL} + N * \text{condTermFlagA}$, donde *M* o *N* pueden ser 1 o 2. Alternativamente, `ctxIdxInc` puede derivarse mediante $\text{ctxIdxInc} = \text{condTermFlagA}$. Alternativamente, `ctxIdxInc` puede derivarse mediante $\text{ctxIdxInc} = \text{condTermFlagL}$. Alternativamente, `ctxIdxInc` puede fijarse en 0.

Como se señaló anteriormente, en algunos ejemplos, los factores de ponderación pueden modificarse. Por ejemplo, la cantidad de factores de ponderación disponibles puede alterarse a nivel de secuencia (por ejemplo, en un conjunto de parámetros, tal como un conjunto de parámetros de secuencia (SPS)). En un ejemplo con fines ilustrativos, un indicador puede señalizarse en un SPS para desactivar uno o más factores de ponderación, por ejemplo, 0,5 y/o 1. En otro ejemplo, dicho indicador puede señalizarse en VPS y aplicarse a todas las vistas sin base. En otro ejemplo más, dicho indicador puede señalizarse en la extensión del conjunto de parámetros de vídeo (VPS) para cada vista sin base. En otro ejemplo, dicho indicador puede proporcionarse en un conjunto de parámetros de imagen (PPS), un encabezado de fragmento o un conjunto de parámetros de vista para deshabilitar uno o más factores de ponderación.

Según otros aspectos, se puede proporcionar un indicador para modificar y/o reemplazar uno o más factores de ponderación. En un ejemplo, el codificador de vídeo puede reemplazar el factor de ponderación de 0,5 por un factor de ponderación de 0,75. Este indicador puede señalizarse en un encabezado de fragmento, un SPS, un conjunto de parámetros de imagen (PPS) o un VPS.

En un ejemplo, un conjunto de parámetros de vídeo se puede modificar de la siguiente manera (por ejemplo, con respecto a la versión 5.0 de 3D-HTM):

Extensión del conjunto de parámetros de vídeo

<code>vps_extension()</code> {	Descriptor
<code>while(!byte_aligned())</code>	
<code> vps_extension_byte_alignment_reserved_one_bit</code>	<code>u(1)</code>
<code>}</code>	
<code>...</code>	
<code>for(i = 0; i <= vps_max_layers_minus1; i++) {</code>	
<code> if (i) {</code>	
<code> multi_view_mv_pred_flag[i]</code>	<code>u(1)</code>
<code> advanced_residual_pred_flag[i]</code>	<code>u(1)</code>
<code> if (advanced_residual_pred_flag[i]) {</code>	
<code> weight_factor_change_flag[i]</code>	<code>u(1)</code>
<code> if(weight_factor_change_flag[i]) {</code>	
<code> diff_weight[i]</code>	<code>se(v)</code>

(continuación)

vps_extension() {	Descriptor
}	
}	
if (i%1) {	
enable_dmm_flag[i]	u(1)
use_mvi_flag[i]	u(1)
}	
}	
}	

En el ejemplo anterior, advanced_residual_pred_flag[i] igual a 1 puede especificar que la predicción residual avanzada (ARP) se puede utilizar para la vista de textura actual con id de capa igual a i. advanced_residual_pred_flag[i] igual a 0 especifica que ARP no se utiliza para la vista de textura actual con id de capa igual a i. Cuando no está presente, se puede inferir que advanced_residual_pred_flag[i] es igual a 0.

En otro ejemplo, una bandera, a saber, advanced_residual_pred_flag puede señalizarse una vez en la extensión de VPS y puede ser aplicable a todas las vistas de textura sin base. En este ejemplo, weight_factor_change_flag[i] igual a 1 puede especificar que el factor de ponderación correspondiente al índice de factor de ponderación igual a 2 se cambie para la capa actual. Además, weight_factor_change_flag[i] igual a 0 puede especificar que el factor de ponderación correspondiente al índice de factor de ponderación igual a 2 no cambie para la capa actual. Además, diff_weight[i] puede especificar una diferencia (con posible escalado) entre el nuevo factor de ponderación y el factor de ponderación original para un índice de factor de ponderación igual a 2. El intervalo de diff_weight[i] puede ser de -2 a 4, inclusive.

En el ejemplo anterior, el codificador de vídeo puede derivar el nuevo factor de ponderación de la siguiente manera:

$$W_2 = (W_2 * 4 + \text{diff_weight}[i]) \div 4.$$

En el ejemplo anterior, cuando el factor de ponderación W_2 es igual a W_0 o W_1 , el índice del factor de ponderación de cualquier CU en la vista aplicable siempre es menor que 2.

En otro ejemplo más, los elementos sintácticos descritos anteriormente pueden señalizarse en el conjunto de parámetros de secuencia o la extensión del conjunto de parámetros de secuencia como advanced_residual_pred_flag, weight factor change_flag y diff_weight para lograr la misma funcionalidad para una vista de textura sin base que se refiere al conjunto de parámetros de secuencia.

La Figura 11 ilustra ubicaciones de muestra en datos de vídeo. En general, las ubicaciones de muestra pueden identificarse mediante un vector de movimiento o vector de disparidad en la codificación de vídeo. Un codificador de vídeo (tal como el codificador de vídeo 20 y/o el decodificador de vídeo 30) puede usar las muestras asociadas con la ubicación identificada con fines de codificación predictiva. En el ejemplo de la Figura 11, las muestras enteras se indican con letras mayúsculas, mientras que las posiciones de las muestras fraccionarias se indican con letras minúsculas. En el ejemplo de la Figura 11 ilustra generalmente la interpolación de luma de un cuarto de muestra, se puede aplicar una interpolación similar para los componentes de croma.

Cuando un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) realiza ARP para un PU, el codificador de vídeo puede necesitar acceder a tres bloques (es decir, B_r , B_c y D_r en la Figura 9). Como se señaló anteriormente, si un vector de movimiento indica una ubicación de pel fraccional, el codificador de vídeo realiza dos procedimientos de interpolación de pel fraccional, por ejemplo, un procedimiento de interpolación para localizar el bloque de referencia temporal y otro procedimiento de interpolación para localizar el bloque de referencia de disparidad-temporal. Además, el codificador de vídeo puede aplicar otro procedimiento de interpolación de pel fraccional al determinar un bloque de referencia de disparidad. HEVC puede usar un filtro de interpolación de luma/croma de 8/4 tomas para un procedimiento de interpolación de muestra fraccional al determinar bloques compensados por movimiento.

De acuerdo con aspectos de esta descripción, el procedimiento de compensación de movimiento de ARP puede simplificarse, particularmente con respecto a la interpolación de sub-pel de bloques de referencia. En algunos casos, de acuerdo con aspectos de esta divulgación, el codificador de vídeo puede usar uno o más tipos de interpolaciones

para determinar las ubicaciones de los bloques de referencia en ARP. Por ejemplo, el codificador de vídeo puede utilizar un filtro de paso bajo, tal como un filtro bilineal, para interpolar la ubicación de los bloques de referencia. En general, un filtro bilineal (es decir, interpolación bilineal) es una extensión de la interpolación lineal para interpolar funciones de dos variables (por ejemplo, x e y) en una cuadrícula bidimensional regular. Por lo tanto, un filtro bilineal puede ser un filtro de 2 tomas.

En algunos ejemplos, el codificador de vídeo puede usar un filtro bilineal al generar el bloque de referencia de disparidad y el bloque de referencia de disparidad temporal. Por consiguiente, el filtro de interpolación de luma/croma de 8/4 tomas utilizado en HEVC para el procedimiento de interpolación de muestras fraccionarias puede reemplazarse por un filtro bilineal al generar el predictor residual, es decir, generar el B_r y B_c que se muestran en la Figura 9.

Además, en algunos ejemplos, el codificador de vídeo puede usar un filtro bilineal cuando genera un bloque de movimiento compensado del PU actual. Es decir, el filtro de interpolación de luma/croma de 8/4 tomas utilizado en HEVC para el procedimiento de interpolación de muestras fraccionarias puede reemplazarse por un filtro bi-lineal cuando se genera el bloque de movimiento compensado del PU actual, es decir, generar el D_r que se muestra en la Figura 9. Por lo tanto, al determinar un bloque predictivo para un PU actual, el codificador de vídeo puede aplicar un filtro bilineal a los componentes de luma y/o croma de la imagen de referencia temporal.

En un ejemplo alternativo, el codificador de vídeo puede aplicar el filtro bilineal descrito anteriormente solo a la luminancia o solo al componente de crominancia. En otro ejemplo, el codificador de vídeo puede aplicar el filtro bilineal a los componentes de luminancia y crominancia.

En el ejemplo que se muestra en la Figura 11, las entradas al procedimiento de interpolación de muestras de luma pueden incluir una ubicación de luma en conjuntos de muestra completa (x_{Int_L} , y_{Int_L}), una ubicación de luma en conjuntos de muestra fraccionaria (x_{Frac_L} , y_{Frac_L}) y la matriz de muestras de referencia de luma $refPicLX_L$. Además, la salida del procedimiento de interpolación es un valor de muestra de luma $predSampleLX_L[x_L, y_L]$

Las posiciones etiquetadas con letras mayúsculas A_i dentro de los bloques sombreados representan muestras de luminancia en ubicaciones de muestra completa dentro de la matriz bidimensional dada $refPicLX_L$ de muestras de luminancia. Estas muestras se pueden utilizar para generar el valor de muestra de luma $predSampleLX_L[x_L, y_L]$. Las ubicaciones (x_{A_i} , y_{A_i}) para cada una de las muestras de luminancia correspondientes A_i dentro de la matriz dada $refPicLX_L$ de muestras de luminancia se pueden derivar de la siguiente manera:

$$x_{A_{i,j}} = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x_{Int_L} + i) \quad (6-1)$$

$$y_{A_{i,j}} = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y_{Int_L} + j) \quad (6-2)$$

Las posiciones etiquetadas con letras minúsculas dentro de bloques sin sombrear representan muestras de luma en ubicaciones fraccionarias de muestras de un cuarto de píxel (un cuarto de pel). El desplazamiento de ubicación de luminancia en conjuntos de muestra fraccionaria (x_{Frac_L} , y_{Frac_L}) especifica cuál de las muestras de luminancia generadas en ubicaciones de muestra completa y de muestra fraccionaria se asigna al valor de muestra de luminancia $predSampleLX_L[x_L, y_L]$ predicho. Esta asignación se puede realizar de acuerdo con la asignación especificada en la Tabla 6-1, que se muestra a continuación. El valor de $predSampleLX_L[x_L, y_L]$ es la salida.

Las variables $shift1$, $shift2$ y $shift3$ pueden derivarse de la misma manera que la subcláusula 8.5.2.2.2.2 de HEVC. Dadas las muestras de luma A_i en ubicaciones de muestra completa (x_{A_i} , y_{A_i}), las muestras de luma ' a_{00} ' a ' r_{00} ' en posiciones de muestra fraccionarias pueden obtenerse mediante las siguientes ecuaciones.

- Las muestras etiquetadas como $a_{0,0}$, $b_{0,0}$, $c_{0,0}$, $d_{0,0}$, $h_{0,0}$ y $n_{0,0}$ se pueden derivar aplicando un filtro de 2 tomas a las muestras de posición entera más cercanas:

$$a_{0,0} = (48 * A_{0,0} + 16 * A_{1,0}) >> \text{shift1} \quad (6-3)$$

$$b_{0,0} = (32 * A_{0,0} + 32 * A_{1,0}) >> \text{shift1} \quad (6-4)$$

$$c_{0,0} = (16 * A_{0,0} + 48 * A_{1,0}) >> \text{shift1} \quad (6-5)$$

$$d_{0,0} = (48 * A_{0,0} + 16 * A_{0,1}) >> \text{shift1} \quad (6-6)$$

$$h_{0,0} = (32 * A_{0,0} + 32 * A_{0,1}) >> \text{shift1} \quad (6-7)$$

$$n_{0,0} = (16 * A_{0,0} + 48 * A_{0,1}) >> \text{shift1} \quad (6-8)$$

Las muestras etiquetadas $e_{0,0}$, $i_{0,0}$, $p_{0,0}$, $f_{0,0}$, $j_{0,0}$, $q_{0,0}$, $g_{0,0}$, $k_{0,0}$ y $r_{0,0}$ pueden derivarse aplicando el filtro de 8 tomas a las muestras $a_{0,j}$, $b_{0,j}$ y $c_{0,j}$ donde $i = -3..4$ en dirección vertical:

$$e_{0,0} = (48 * a_{0,0} + 16 * a_{0,1}) >> \text{shift2} \quad (6-9)$$

$$i_{0,0} = (32 * a_{0,0} + 32 * a_{0,1}) >> \text{shift2} \quad (6-10)$$

$$p_{0,0} = (16 * a_{0,0} + 48 * a_{0,1}) >> \text{shift2} \quad (6-11)$$

$$f_{0,0} = (48 * b_{0,0} + 16 * b_{0,1}) >> \text{shift2} \quad (6-12)$$

$$j_{0,0} = (32 * b_{0,0} + 32 * b_{0,1}) >> \text{shift2} \quad (6-13)$$

$$q_{0,0} = (16 * b_{0,0} + 48 * b_{0,1}) >> \text{shift2} \quad (6-14)$$

$$g_{0,0} = (48 * c_{0,0} + 16 * c_{0,1}) >> \text{shift2} \quad (6-15)$$

$$k_{0,0} = (32 * c_{0,0} + 32 * c_{0,1}) >> \text{shift2} \quad (6-16)$$

$$r_{0,0} = (16 * c_{0,0} + 48 * c_{0,1}) >> \text{shift2} \quad (6-17)$$

Tabla 6-1 - Asignación de la muestra de predicción de luminancia predSampleLXL[xL, yL]

xFracL	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
yFracL	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
predSampleLXL[xL, yL]	A << shif t3	d	h	n	a	e	i	p	b	f	j	q	c	g	k	r

Como se señaló anteriormente, aunque se describe con respecto a un componente de luma, un codificador de vídeo puede localizar muestras en un bloque de croma de una manera similar.

En algunos ejemplos, las ubicaciones del bloque de referencia de disparidad y el bloque de referencia de disparidad temporal pueden determinarse después de que se haya aplicado la compensación de movimiento como se especifica en la subcláusula 8.5.2.2.1 y 8.5.2.2.2 de la especificación HEVC. Por ejemplo, para el bloque actual, la matriz de muestras de luma predicha puede identificarse como predSam- pleLXL, y las matrices de muestras de croma pueden identificarse como predSampleLXcb y predSampleLXcr. En este ejemplo, si el factor de ponderación no es igual a 0, el codificador de vídeo puede realizar las siguientes operaciones al final del procedimiento:

- Para cada lista de imágenes de referencia X (siendo X 0 o 1), si la imagen de referencia no es una imagen de referencia inter- vista, se aplica lo siguiente para modificar aún más los valores de muestra previstos:

1. Invocó el procedimiento de derivación del vector de disparidad para obtener un vector de disparidad, apuntando a una vista de referencia diana.
2. Localice el bloque de referencia mediante el vector de disparidad en la vista de referencia de destino de la imagen dentro del mismo conjunto de acceso. Si el vector de disparidad apunta a una posición fraccionaria (es decir, la posición superior izquierda del bloque de referencia (B_c en la Figura 9) es fraccional), se aplica un filtro bilineal para interpolar el bloque de referencia.
3. Reutilice la información de movimiento del bloque actual para derivar la información de movimiento para el bloque de referencia. Aplicar compensación de movimiento para el bloque de referencia basado en el vector de movimiento derivado del bloque de referencia y la imagen de referencia derivada en la vista de referencia para el bloque de referencia, para derivar un bloque de residuos. La relación entre el bloque actual, el bloque de referencia y el bloque con compensación de movimiento se muestra en la Figura 9.

- Denotar el índice de referencia del bloque actual como ref_idx_lx
- Seleccionar en el búfer de imágenes decodificadas una imagen de referencia que tenga el mismo POC que refPicListX [ref_idx_lx] y dentro de la vista de referencia diana.

• Derivar el vector de movimiento del bloque de referencia para que sea el mismo que el vector de movimiento del bloque actual. *La interpolación bilineal se aplica si el vector de movimiento apunta a una posición fraccionaria, es decir, la posición superior izquierda del bloque de referencia más el vector de movimiento es una posición fraccionaria (posición superior izquierda de B_r en la Figura 9).*

4. Aplicar el factor de ponderación al bloque de residuos para obtener un bloque de residuos ponderado, denotado como predARPSampleLX_L , $\text{predARPSampleLX}_{cb}$ y $\text{predARPSampleLX}_{cr}$,
5. Agregue los valores del bloque de residuos ponderados a las muestras previstas:

• $\text{predSampleLX}_L = \text{predSampleLX}_L + \text{predARPSampleLX}_L$
• $\text{predSampleLX}_{cb} = \text{predSampleLX}_{cb} + \text{predARPSampleLX}_{cb}$
• $\text{predSampleLX}_{cr} = \text{predSampleLX}_{cr} + \text{predARPSampleLX}_{cr}$

Tenga en cuenta que la operación anterior es una operación de adición de matriz/vector.

Independientemente de si se aplica o no la predicción residual avanzada inter-vista, el procedimiento de predicción de muestra ponderada, tal como se especifica en la subcláusula 8.5.2.2.3 de la especificación HEVC (por ejemplo, WD9), se aplica para un PU predicha bidireccional.

Como se señaló anteriormente, según algunos aspectos de esta divulgación, cuando la o las listas de imágenes de referencia del bloque de referencia de disparidad no incluyen una imagen de referencia en la misma vista que la imagen de referencia de disparidad que tiene el mismo POC que la imagen de referencia temporal, el codificador de vídeo puede modificar el procedimiento ARP.

En algunos ejemplos, el codificador de vídeo puede modificar el procedimiento ARP deshabilitando el procedimiento ARP, de modo que el bloque actual no esté codificado usando ARP. En otros ejemplos, el codificador de vídeo puede modificar el procedimiento ARP escalando el vector de movimiento temporal para identificar otra imagen de referencia de disparidad temporal. Por ejemplo, el codificador de vídeo puede escalar el vector de movimiento temporal de modo que, cuando se aplica a la imagen de referencia de disparidad, el vector de movimiento escalado identifica una imagen de referencia de disparidad temporal que se incluye en la lista de imágenes de referencia y está en una ubicación temporalmente más cercana a la imagen de referencia de disparidad.

La Figura 12 ilustra generalmente modos de partición (que pueden definir tamaños de PU) que pueden estar asociados con conjuntos de predicción. Por ejemplo, suponiendo que el tamaño de un CU particular es $2N \times 2N$, el CU puede predecirse usando los modos de partición $2N \times 2N$ (160), $N \times N$ (162), $hN \times 2N$ (164), $2N \times hN$ (166), $N \times 2N$ (168), $2N \times N$ (170), $nL \times 2N$ (172), $nR \times 2N$ (174), $2N \times nU$ (176) y $2N \times nD$ (178). Los modos de partición mostrados en el ejemplo de la Figura 12 se presentan solo con fines ilustrativos, y se pueden usar otros modos de partición para indicar la manera en que se predicen los datos de vídeo.

En algunos casos, un codificador de vídeo (por ejemplo, tal como el codificador de vídeo 20 y/o el decodificador de vídeo 30) puede realizar intrapredicción o interpredicción utilizando los modos de partición 160 y 162. Por ejemplo, el codificador de vídeo puede predecir un CU como un todo usando el PU $2N \times 2N$ (modo de partición 160). En otro ejemplo, el codificador de vídeo puede predecir el CU utilizando cuatro PU de tamaño $N \times N$ (modo de partición 162), con cada una de las cuatro secciones que tienen una técnica de predicción potencialmente diferente que se aplica.

Además, con respecto a la intracodificación, el codificador de vídeo puede realizar una técnica denominada intrapredicción de corta distancia (SDIP). Si SDIP está disponible, el CU puede predecirse utilizando PU paralelas (modos de partición 164 y 166). Es decir, SDIP generalmente permite que un CU se divida en PU paralelos. Al dividir un conjunto de codificación (CU) en conjuntos de predicción no cuadrados (PU), las distancias entre los píxeles predichos y de referencia se pueden acortar.

Con respecto a la intercodificación, además de los modos de partición simétrica 160 y 162, el codificador de vídeo puede implementar una disposición lado a lado de PU (modos de partición 168 y 170), o una variedad de modos AMP (partición de movimiento asimétrico). Con respecto a los modos AMP, el codificador de vídeo puede dividir asimétricamente un CU utilizando los modos de partición $nL \times 2N$ (172), $nR \times 2N$ (174), $2N \times nU$ (176) y $2N \times nD$ (178). En la partición asimétrica, una dirección de un CU no se divide, mientras que la otra dirección se divide en 25 % y 75 %. La porción del CU correspondiente a la partición del 25 % se indica con una "n" seguida de una indicación de "Arriba", "Abajo", "Izquierda" o "Derecha".

De acuerdo con otros aspectos de esta descripción, ARP puede habilitarse o deshabilitarse en función de un modo de partición y/o modo de codificación del bloque que se está codificando actualmente. Por ejemplo, los factores de ponderación solo pueden señalizarse para ciertos modos de partición y/o ciertos modos de codificación. Si un factor de ponderación no se incluye en un flujo de bits, un decodificador de vídeo puede omitir la decodificación del factor de ponderación e inferir que el factor de ponderación es de valor cero (desactivando así ARP).

En un ejemplo, como se señaló anteriormente con respecto a las tablas sintácticas de conjuntos de codificación de ejemplo, de acuerdo con algunos aspectos de esta descripción, el factor de ponderación para cualquier bloque intercodificado con modo de partición desigual a PART_2Nx2N (modo de partición 160) puede no señalizarse. En otro ejemplo, el factor de ponderación para un bloque intercodificado con un modo de partición distinto de PART_2Nx2N (modo de partición 160), PART_2NxN (modo de partición 170) y PART_Nx2N (modo de partición 168) puede no ser señalado. En incluso otro ejemplo, adicional o alternativamente, el factor de ponderación para cualquier bloque intercodificado con modo de codificación desigual al modo de salto y/o modo de fusión puede no señalizarse.

La Figura 13 es un diagrama de flujo que ilustra un procedimiento ejemplar para codificar un bloque actual de acuerdo con las técnicas de la presente divulgación. El bloque actual puede comprender un CU actual o una porción del CU actual, por ejemplo, un PU actual. Aunque se describe con respecto al codificador de vídeo 20 (Figuras 1 y 2), debe entenderse que otros dispositivos pueden configurarse para realizar un procedimiento similar al de la Figura 13.

En este ejemplo, el codificador de vídeo 20 predice inicialmente el bloque actual (190) utilizando un vector de movimiento. Por ejemplo, el codificador de vídeo 20 puede calcular uno o más conjuntos de predicción (PU) para el bloque actual. En este ejemplo, se supone que el codificador de vídeo 20 interpredice el bloque actual. Por ejemplo, el conjunto de estimación de movimiento 42 puede calcular un vector de movimiento para el bloque actual realizando una búsqueda de movimiento de imágenes codificadas previamente, por ejemplo, imágenes entre visualizaciones e imágenes temporales. Por lo tanto, el conjunto de estimación de movimiento 42 puede producir un vector de movimiento temporal o un vector de movimiento de disparidad para codificar el bloque actual.

El codificador de vídeo 20 puede entonces determinar si las listas de imágenes de referencia (por ejemplo, RefPicList0 y RefPicList 1, cuando el bloque actual se bipredice) para codificar el bloque actual incluyen una o más imágenes de referencia en una ubicación temporal que no sea la ubicación temporal del bloque actual (191). En algunos ejemplos, el codificador de vídeo 20 puede hacer dicha determinación determinando si el bloque actual está incluido en una imagen de acceso aleatorio, como se describe en otra parte de esta divulgación.

Si las listas de imágenes de referencia incluyen una imagen de referencia en una ubicación temporal que es diferente de la ubicación temporal del bloque actual (la rama sí de la etapa 191), el codificador de vídeo 20 puede habilitar un procedimiento de predicción residual inter-visualización, tal como el procedimiento ARP descrito anteriormente. En este ejemplo, el codificador de vídeo 20 puede realizar una predicción interresidual para predecir los datos residuales del bloque actual (192). Por ejemplo, como se señaló anteriormente, el codificador de vídeo 20 puede determinar un bloque de referencia de disparidad indicado por un vector de disparidad del primer bloque, determinar un bloque de referencia de temporaldisparidad combinando el vector de movimiento temporal y el vector de movimiento de disparidad, y determinar un predictor residual en función de una diferencia entre el bloque de referencia de disparidad temporal y el bloque de referencia de disparidad. El codificador de vídeo 20 puede aplicar un factor de ponderación al predictor residual. El codificador de vídeo 20 puede entonces calcular un bloque residual para el bloque actual (194).

Si las listas de imágenes de referencia no incluyen una imagen de referencia en una ubicación temporal que es diferente de la ubicación temporal del bloque actual (la rama no de la etapa 191), el codificador de vídeo 20 puede deshabilitar un procedimiento de predicción residual inter-visualización, tal como el procedimiento ARP descrito anteriormente, y puede omitir el cálculo del bloque residual para el bloque actual (194). En este ejemplo, el codificador de vídeo 20 puede no señalizar un factor de ponderación para el procedimiento de predicción residual inter-visualización. Es decir, en un ejemplo con fines ilustrativos, el codificador de vídeo 20 puede no señalizar un elemento sintáctico `weighting_factor_index` en el flujo de bits.

En cualquier caso, el codificador de vídeo 20 calcula el bloque residual para el bloque actual, por ejemplo, para producir un conjunto de transformación (TU) (194). Para calcular el bloque residual cuando no se utiliza la predicción residual inter-visualización, el codificador de vídeo 20 puede calcular una diferencia entre el bloque original no codificado y el bloque predicho para que el bloque actual produzca el residuo. Para calcular el bloque residual cuando se utiliza la predicción residual inter-visualización, el codificador de vídeo 20 puede calcular una diferencia entre el bloque original no codificado y el bloque predicho para que el bloque actual genere un primer residuo. El codificador de vídeo 20 puede entonces calcular un residuo final en función de la diferencia entre el primer residuo y el predictor residual.

El codificador de vídeo 20 puede entonces transformar y cuantificar los coeficientes del bloque residual (196). A continuación, el codificador de vídeo 20 puede escanear los coeficientes de transformación cuantificados del bloque residual (198). Durante el escaneo, o después del escaneo, el codificador de vídeo 20 puede codificar por entropía los coeficientes de transformación que incluyen, por ejemplo, un valor de ponderación de predicción residual inter-visualización en casos en los que la predicción residual inter-visualización está habilitada y aplicada (200). El codificador de vídeo 20 puede entonces emitir los datos codificados por entropía para los coeficientes del bloque y el valor de ponderación en los casos en los que la predicción residual inter-visualización está habilitada y aplicada (202).

La Figura 14 es un diagrama de flujo que ilustra un procedimiento ejemplar para decodificar un bloque actual de datos de vídeo de acuerdo con las técnicas de la presente divulgación. El bloque actual puede comprender un CU actual o una porción del CU actual (por ejemplo, un PU). Aunque se describe con respecto al decodificador de vídeo 30 (Figuras

1 y 3), debe entenderse que otros dispositivos pueden configurarse para realizar un procedimiento similar al de la Figura 14.

Inicialmente, el decodificador de vídeo 30 recibe datos para los coeficientes de transformación y un vector de movimiento para el bloque actual (210). De nuevo, este ejemplo supone que el bloque actual está interpretado. El conjunto de decodificación entrópica 80 decodifica entrópicamente los datos para los coeficientes y el vector de movimiento para el bloque (212).

El decodificador de vídeo 30 puede determinar si las listas de imágenes de referencia (por ejemplo, RefPicListO y RefPicList 1, cuando el bloque actual está bipredicho) para codificar el bloque actual incluyen una o más imágenes de referencia en una ubicación temporal que no sea la ubicación temporal del bloque actual (214). En algunos ejemplos, el decodificador de vídeo 30 puede realizar dicha determinación determinando si el bloque actual está incluido en una imagen de acceso aleatorio, como se describe en otra parte de esta divulgación.

Si las listas de imágenes de referencia incluyen una imagen de referencia en una ubicación temporal que es diferente de la ubicación temporal del bloque actual (la rama sí de la etapa 214), el decodificador de vídeo 30 puede habilitar un procedimiento de predicción residual inter-vista, tal como el procedimiento ARP descrito anteriormente. En este ejemplo, el decodificador de vídeo 30 puede realizar predicción interresidual para predecir los datos residuales del bloque actual (216). Por ejemplo, como se señaló anteriormente, el decodificador de vídeo 30 puede determinar un bloque de referencia de disparidad indicado por un vector de disparidad del primer bloque, determinar un bloque de referencia de disparidad temporal combinando el vector de movimiento temporal y el vector de movimiento de disparidad, y determinar un predictor residual en función de una diferencia entre el bloque de referencia de disparidad temporal y el bloque de referencia de disparidad. El decodificador de vídeo 30 también puede aplicar un factor de ponderación, como se señala en el flujo de bits, al predictor residual.

Si las listas de imágenes de referencia no incluyen una imagen de referencia en una ubicación temporal que es diferente de la ubicación temporal del bloque actual (la rama no de la etapa 214), o después de predecir datos residuales con predicción residual inter-vista (216)), el decodificador de vídeo 30 puede deshabilitar un procedimiento de predicción residual inter-vista, tal como el procedimiento ARP descrito anteriormente, y puede omitir la predicción del bloque actual utilizando el vector de movimiento (218).

En cualquier caso, el decodificador de vídeo 30 puede predecir el bloque actual utilizando el vector de movimiento decodificado (218). El decodificador de vídeo 30 puede entonces escanear inversamente los coeficientes reproducidos (220), para crear un bloque de coeficientes de transformación cuantificados. El decodificador de vídeo 30 puede entonces cuantificar inversamente y transformar inversamente los coeficientes para producir un bloque residual (222). El decodificador de vídeo 30 puede decodificar en última instancia el bloque actual combinando el bloque predicho y el o los bloques residuales (224). Por ejemplo, en casos en los que no se aplica la predicción residual inter-vista, el decodificador de vídeo 30 puede combinar simplemente el bloque predicho y el residual decodificado. En los casos en los que se aplica la predicción residual inter-vista, el decodificador de vídeo 30 puede combinar el bloque predicho, el residual decodificado (que representa un residual final) y el predictor residual.

La Figura 15 es un diagrama de flujo que ilustra un procedimiento ejemplar para codificar un bloque actual de acuerdo con las técnicas de la presente divulgación. El bloque actual puede comprender un CU actual o una porción del CU actual, por ejemplo, un PU actual. Aunque se describe con respecto al codificador de vídeo 20 (Figuras 1 y 2), debe entenderse que otros dispositivos pueden configurarse para realizar un procedimiento similar al de la Figura 15.

En este ejemplo, el codificador de vídeo 20 determina la ubicación de un bloque de referencia temporal indicado por un vector de movimiento temporal para el bloque actual (240). Por ejemplo, el codificador de vídeo 20 puede calcular uno o más conjuntos de predicción (PU) para el bloque actual. En este ejemplo, se supone que el codificador de vídeo 20 interpreta el bloque actual. Por ejemplo, el conjunto de estimación de movimiento 42 puede calcular un vector de movimiento para el bloque actual realizando una búsqueda de movimiento de imágenes codificadas previamente, por ejemplo, imágenes entre visualizaciones e imágenes temporales. Por lo tanto, el conjunto de estimación de movimiento 42 puede producir un vector de movimiento temporal o un vector de movimiento de disparidad para codificar el bloque actual.

El codificador de vídeo 20 también puede interpolar la ubicación de un bloque de referencia de disparidad (242). Por ejemplo, el codificador de vídeo 20 puede determinar un vector de disparidad para ubicar un bloque de referencia de disparidad que tiene el mismo valor de POC que el bloque actual, pero ubicado en una segunda vista diferente. En algunos casos, de acuerdo con aspectos de esta divulgación, si el vector de disparidad identifica una ubicación para el bloque de referencia de disparidad que no es una ubicación entera, el codificador de vídeo 20 puede aplicar un filtro bilineal para interpolar la ubicación del bloque de referencia de disparidad.

Además, el codificador de vídeo 20 puede determinar la ubicación de un bloque de referencia de disparidad temporal (244). Por ejemplo, el codificador de vídeo 20 puede combinar el vector de movimiento temporal y el vector de movimiento de disparidad para determinar la ubicación del bloque de referencia de disparidad temporal. De nuevo, en

algunos casos, de acuerdo con aspectos de esta divulgación, si la combinación identifica una ubicación para el bloque de referencia de disparidad temporal que no es una ubicación entera, el codificador de vídeo 20 puede aplicar un filtro bilineal para interpolar la ubicación del bloque de referencia de disparidad temporal.

5 El codificador de vídeo 20 puede entonces determinar un predictor residual para el bloque actual (246). El codificador de vídeo 20 puede determinar el predictor residual en función de la diferencia entre el bloque de referencia de disparidad y el bloque de referencia de disparidad temporal. El codificador de vídeo 20 puede aplicar un factor de ponderación al predictor residual resultante.

10 El codificador de vídeo 20 puede entonces determinar un residuo final para el bloque (248). Por ejemplo, el codificador de vídeo 20 puede determinar un primer residuo en función de una diferencia entre las muestras del bloque actual y el bloque de referencia temporal. El codificador de vídeo 20 puede entonces determinar el residuo final en función de una diferencia entre el primer residuo y el predictor residual.

15 El codificador de vídeo 20 puede entonces transformar y cuantificar los coeficientes del bloque residual (250). A continuación, el codificador de vídeo 20 puede escanear los coeficientes de transformación cuantificados del bloque residual (252). Durante el escaneo, o después del escaneo, el codificador de vídeo 20 puede codificar por entropía los coeficientes de transformación que incluyen, por ejemplo, un valor de ponderación de predicción residual inter-vista (254). El codificador de vídeo 20 puede entonces emitir los datos codificados por entropía para los coeficientes del bloque y el valor de ponderación (256).

25 La Figura 16 es un diagrama de flujo que ilustra un procedimiento ejemplar para decodificar un bloque actual de datos de vídeo de acuerdo con las técnicas de la presente divulgación. El bloque actual puede comprender un CU actual o una porción del CU actual (por ejemplo, un PU). Aunque se describe con respecto al decodificador de vídeo 30 (Figuras 1 y 3), debe entenderse que otros dispositivos pueden configurarse para realizar un procedimiento similar al de la Figura 14.

30 Inicialmente, el decodificador de vídeo 30 recibe datos para los coeficientes de transformación y un vector de movimiento para el bloque actual (260). De nuevo, este ejemplo supone que el bloque actual está interpredicho. El conjunto de decodificación entrópica 80 decodifica entrópicamente los datos para los coeficientes y el vector de movimiento para el bloque (262).

35 El decodificador de vídeo 30 puede predecir el bloque actual utilizando el vector de movimiento decodificado (264). El decodificador de vídeo 30 también puede escanear inversamente los coeficientes reproducidos (266), para crear un bloque de coeficientes de transformación cuantificados. El decodificador de vídeo 30 también puede cuantificar inversamente y transformar inversamente los coeficientes para producir un bloque residual (268).

40 El decodificador de vídeo 30 también puede interpolar la ubicación de un bloque de referencia de disparidad (270). Por ejemplo, el decodificador de vídeo 30 puede determinar un vector de disparidad para ubicar un bloque de referencia de disparidad que tiene el mismo valor de POC que el bloque actual, pero ubicado en una segunda vista diferente. En algunos casos, de acuerdo con aspectos de esta divulgación, si el vector de disparidad identifica una ubicación para el bloque de referencia de disparidad que no es una ubicación entera, el decodificador de vídeo 30 puede aplicar un filtro bilineal para interpolar la ubicación del bloque de referencia de disparidad.

45 Además, el decodificador de vídeo 30 puede determinar la ubicación de un bloque de referencia de disparidad temporal (272). Por ejemplo, el decodificador de vídeo 30 puede combinar el vector de movimiento temporal y el vector de movimiento de disparidad para determinar la ubicación del bloque de referencia de disparidad temporal. De nuevo, en algunos casos, de acuerdo con aspectos de esta divulgación, si la combinación identifica una ubicación para el bloque de referencia de disparidad temporal que no es una ubicación entera, el decodificador de vídeo 30 puede aplicar un filtro bilineal para interpolar la ubicación del bloque de referencia de disparidad temporal.

50 El decodificador de vídeo 30 puede entonces determinar un predictor residual para el bloque actual (274). El decodificador de vídeo 30 puede determinar el predictor residual en función de la diferencia entre el bloque de referencia de disparidad y el bloque de referencia de disparidad temporal. El decodificador de vídeo 30 puede aplicar un factor de ponderación al predictor residual resultante.

55 El decodificador de vídeo 30 puede decodificar en última instancia el bloque actual combinando el bloque predicho y el residual (276). Por ejemplo, el decodificador de vídeo 30 puede combinar el bloque predicho, el residuo decodificado (que representa un residuo final) y el predictor residual.

60 La Figura 17 es un diagrama de flujo que ilustra un procedimiento ejemplar para codificar un bloque actual de acuerdo con las técnicas de la presente divulgación. El bloque actual puede comprender un CU actual o una porción del CU actual, por ejemplo, un PU actual. Aunque se describe con respecto al codificador de vídeo 20 (Figuras 1 y 2), debe entenderse que otros dispositivos pueden configurarse para realizar un procedimiento similar al de la Figura 17.

65

En este ejemplo, el codificador de vídeo 20 determina inicialmente un modo de partición para predecir el bloque actual (280). Por ejemplo, el codificador de vídeo 20 puede determinar si calcular un PU (por ejemplo, un modo de partición $2N \times 2N$) o calcular más de un PU para el bloque actual. En este ejemplo, se supone que el codificador de vídeo 20 interpredice el bloque actual. Por ejemplo, el conjunto de estimación de movimiento 42 puede calcular un vector de movimiento para el bloque actual realizando una búsqueda de movimiento de imágenes codificadas previamente, por ejemplo, imágenes entre visualizaciones e imágenes temporales. Por lo tanto, el conjunto de estimación de movimiento 42 puede producir un vector de movimiento temporal o un vector de movimiento de disparidad para codificar el bloque actual.

El codificador de vídeo 20 puede entonces determinar si codificar datos que indican un factor de ponderación (y realizar predicción residual de inter-vista) en función del modo de partición determinado (282). En algunos ejemplos, el codificador de vídeo 20 puede deshabilitar la predicción residual inter-vista y omitir la codificación de un factor de ponderación si el modo de partición es un modo distinto del modo de partición $2N \times 2N$.

Si el codificador de vídeo 20 codifica un factor de ponderación, el codificador de vídeo 20 puede realizar una predicción residual inter-vista para predecir los datos residuales del bloque actual (284). Por ejemplo, como se señaló anteriormente, el codificador de vídeo 20 puede determinar un bloque de referencia de disparidad indicado por un vector de disparidad del primer bloque, determinar un bloque de referencia de disparidad temporal combinando el vector de movimiento temporal y el vector de movimiento de disparidad, y determinar un predictor residual en función de una diferencia entre el bloque de referencia de disparidad temporal y el bloque de referencia de disparidad. El codificador de vídeo 20 puede aplicar un factor de ponderación al predictor residual. El codificador de vídeo 20 puede entonces calcular un bloque residual para el bloque actual (286).

Si el codificador de vídeo 20 no codifica un factor de ponderación, (la rama no de la etapa 282) el codificador de vídeo 20 puede deshabilitar la predicción residual inter-vista y puede pasar a calcular el bloque residual para el bloque actual (286). En este ejemplo, el codificador de vídeo 20 puede no señalar un factor de ponderación para el procedimiento de predicción residual inter-vista. Es decir, en un ejemplo con fines ilustrativos, el codificador de vídeo 20 puede no señalar un elemento sintáctico `weighting_factor_index` en el flujo de bits.

En cualquier caso, el codificador de vídeo 20 calcula el bloque residual para el bloque actual, por ejemplo, para producir un conjunto de transformación (TU) (286). Para calcular el bloque residual cuando no se utiliza la predicción residual inter-vista, el codificador de vídeo 20 puede calcular una diferencia entre el bloque original no codificado y el bloque predicho para que el bloque actual produzca el residuo. Para calcular el bloque residual cuando se utiliza la predicción residual inter-vista, el codificador de vídeo 20 puede calcular una diferencia entre el bloque original no codificado y el bloque predicho para que el bloque actual genere un primer residuo. El codificador de vídeo 20 puede entonces calcular un residuo final en función de la diferencia entre el primer residuo y el predictor residual.

El codificador de vídeo 20 puede entonces transformar y cuantificar los coeficientes del bloque residual (288). A continuación, el codificador de vídeo 20 puede escanear los coeficientes de transformación cuantificados del bloque residual (290). Durante el escaneo, o después del escaneo, el codificador de vídeo 20 puede codificar por entropía los coeficientes de transformación que incluyen, por ejemplo, un valor de ponderación de predicción residual inter-vista en casos en los que la predicción residual inter-vista está habilitada y aplicada (292). El codificador de vídeo 20 puede entonces emitir los datos codificados por entropía para los coeficientes del bloque y el valor de ponderación en los casos en los que la predicción residual inter-vista está habilitada y aplicada (294).

La Figura 18 es un diagrama de flujo que ilustra un procedimiento ejemplar para decodificar un bloque actual de datos de vídeo de acuerdo con las técnicas de la presente divulgación. El bloque actual puede comprender un CU actual o una porción del CU actual (por ejemplo, un PU). Aunque se describe con respecto al decodificador de vídeo 30 (Figuras 1 y 3), debe entenderse que otros dispositivos pueden configurarse para realizar un procedimiento similar al de la Figura 14.

En este ejemplo, el decodificador de vídeo 30 determina inicialmente un modo de partición para predecir el bloque actual (300). Por ejemplo, el decodificador de vídeo 30 puede determinar si determinar un PU (por ejemplo, un modo de partición $2N \times 2N$) o determinar más de un PU para el bloque actual. Esa estructura de partición para el bloque puede señalizarse en el flujo de bits codificado. El decodificador de vídeo 30 también decodifica por entropía datos para coeficientes de transformación y un vector de movimiento para el bloque actual (302). De nuevo, este ejemplo supone que el bloque actual está interpredicho.

El decodificador de vídeo 30 puede entonces determinar si decodificar (por ejemplo, analizar, a partir del flujo de bits codificado) un factor de ponderación (y realizar la predicción residual inter-vista) en función del modo de partición determinado (304). En algunos ejemplos, el decodificador de vídeo 20 puede deshabilitar la predicción residual inter-vista y omitir la decodificación de un factor de ponderación si el modo de partición es un modo distinto del modo de partición $2N \times 2N$. Es decir, por ejemplo, el decodificador de vídeo 30 puede determinar automáticamente (es decir, inferir) que el factor de ponderación es cero cuando el modo de partición es un modo distinto del modo de partición $2N \times 2N$.

Si el decodificador de vídeo 30 decodifica un factor de ponderación, el decodificador de vídeo 30 puede realizar una predicción residual inter-vista para predecir los datos residuales del bloque actual (306). Por ejemplo, como se señaló anteriormente, el decodificador de vídeo 30 puede determinar un bloque de referencia de disparidad indicado por un vector de disparidad del primer bloque, determinar un bloque de referencia de disparidad temporal aplicando el vector de movimiento del bloque actual al bloque de referencia de disparidad y determinar un predictor residual en función de una diferencia entre el bloque de referencia de disparidad temporal y el bloque de referencia de disparidad. El decodificador de vídeo 30 también puede aplicar un factor de ponderación, como se señala en el flujo de bits, al predictor residual.

Si el decodificador de vídeo 30 no decodifica un factor de ponderación (la rama no de la etapa 304), el decodificador de vídeo 30 puede deshabilitar un procedimiento de predicción residual inter-vista. El decodificador de vídeo 30 puede pasar a predecir el bloque actual utilizando el vector de movimiento.

En cualquier caso, el decodificador de vídeo 30 puede predecir el bloque actual utilizando el vector de movimiento decodificado (308). El decodificador de vídeo 30 puede entonces escanear inversamente los coeficientes reproducidos (310), para crear un bloque de coeficientes de transformación cuantificados. El decodificador de vídeo 30 puede entonces cuantificar inversamente y transformar inversamente los coeficientes para producir un bloque residual (312). El decodificador de vídeo 30 puede decodificar en última instancia el bloque actual combinando el bloque predicho y el o los bloques residuales (314). Por ejemplo, en casos en los que no se aplica la predicción residual inter-vista, el decodificador de vídeo 30 puede combinar simplemente el bloque predicho y el residual decodificado. En los casos en los que se aplica la predicción residual inter-vista, el decodificador de vídeo 30 puede combinar el bloque predicho, el residual decodificado (que representa un residual final) y el predictor residual.

La Figura 19 es un diagrama de flujo que ilustra un procedimiento ejemplar para codificar un bloque actual de acuerdo con las técnicas de la presente divulgación. El bloque actual puede comprender un CU actual o una porción del CU actual, por ejemplo, un PU actual. Aunque se describe con respecto al codificador de vídeo 20 (Figuras 1 y 2), debe entenderse que otros dispositivos pueden configurarse para realizar un procedimiento similar al de la Figura 19.

En este ejemplo, el codificador de vídeo 20 determina un vector de movimiento temporal y una imagen de referencia para el bloque actual (320). Por ejemplo, el codificador de vídeo 20 puede calcular uno o más conjuntos de predicción (PU) para el bloque actual. En este ejemplo, se supone que el codificador de vídeo 20 interpreta el bloque actual. Por ejemplo, el conjunto de estimación de movimiento 42 puede calcular un vector de movimiento para el bloque actual realizando una búsqueda de movimiento de imágenes codificadas previamente, por ejemplo, imágenes entre visualizaciones e imágenes temporales. Por lo tanto, el conjunto de estimación de movimiento 42 puede producir un vector de movimiento temporal o un vector de movimiento de disparidad para codificar el bloque actual.

El codificador de vídeo 20 puede entonces determinar un bloque de referencia de disparidad en el mismo conjunto de acceso que el bloque actual (322). Por ejemplo, el codificador de vídeo 20 puede determinar un vector de disparidad para ubicar un bloque de referencia de disparidad que tiene el mismo valor de POC que el bloque actual, pero ubicado en una segunda vista diferente.

El codificador de vídeo 20 puede determinar si un búfer de imágenes decodificadas (también denominada memoria de imágenes de referencia en esta invención) contiene una imagen que tiene un valor de POC que es igual al valor de POC de la imagen de referencia temporal (324). Por ejemplo, el codificador de vídeo 20 puede determinar si la imagen indicada por una combinación del vector de movimiento temporal y el vector de movimiento de disparidad está incluida en el búfer de imágenes decodificadas. En algunos casos, incluso si la imagen de referencia de disparidad temporal potencial está incluida en el búfer de imágenes decodificadas, el codificador de vídeo 20 puede determinar además si la imagen está incluida en una o ambas listas de imágenes de referencia para el bloque de referencia de disparidad.

Si la imagen de referencia de disparidad temporal potencial está incluida en el búfer de imágenes decodificadas (y/o una o ambas listas de imágenes de referencia del bloque de referencia de disparidad) (324), el codificador de vídeo 20 puede realizar un procedimiento de predicción residual inter-vista para predecir los datos residuales del bloque actual (326). Por ejemplo, como se señaló anteriormente, el codificador de vídeo 20 puede determinar un bloque de referencia de disparidad indicado por un vector de disparidad del primer bloque, determinar un bloque de referencia de disparidad temporal aplicando el vector de movimiento del bloque actual al bloque de referencia de disparidad y determinar un predictor residual en función de una diferencia entre el bloque de referencia de disparidad temporal y el bloque de referencia de disparidad. El codificador de vídeo 20 puede aplicar un factor de ponderación al predictor residual. El codificador de vídeo 20 puede entonces calcular un bloque residual para el bloque actual (330).

Si la imagen de referencia de disparidad temporal potencial no está incluida en el búfer de imágenes decodificadas (o no está incluida en una o ambas listas de imágenes de referencia del bloque de referencia de disparidad) (la rama no de la etapa 324), el codificador de vídeo 20 puede modificar el procedimiento de predicción residual inter-vista (328). En algunos ejemplos, el codificador de vídeo 20 puede modificar el procedimiento deshabilitando el procedimiento. En otros ejemplos, el codificador de vídeo 20 puede seleccionar una imagen de referencia disponible (una imagen de

referencia que se incluye en el búfer de imágenes decodificadas y/o la lista de imágenes de referencia) y escalar el vector de movimiento temporal en consecuencia.

En cualquier caso, el codificador de vídeo 20 calcula el bloque residual para el bloque actual, por ejemplo, para producir un conjunto de transformación (TU) (330). Para calcular el bloque residual cuando no se utiliza la predicción residual inter-vista, el codificador de vídeo 20 puede calcular una diferencia entre el bloque original no codificado y el bloque predicho para que el bloque actual produzca el residuo. Para calcular el bloque residual cuando se utiliza la predicción residual inter-vista, el codificador de vídeo 20 puede calcular una diferencia entre el bloque original no codificado y el bloque predicho para que el bloque actual genere un primer residuo. El codificador de vídeo 20 puede entonces calcular un residuo final en función de la diferencia entre el primer residuo y el predictor residual.

El codificador de vídeo 20 puede entonces transformar y cuantificar los coeficientes del bloque residual (332). A continuación, el codificador de vídeo 20 puede escanear los coeficientes de transformación cuantificados del bloque residual (334). Durante el escaneo, o después del escaneo, el codificador de vídeo 20 puede codificar por entropía los coeficientes de transformación que incluyen, por ejemplo, un valor de ponderación de predicción residual inter-vista en casos en los que la predicción residual inter-vista está habilitada y aplicada (336). El codificador de vídeo 20 puede entonces emitir los datos codificados por entropía para los coeficientes del bloque y el valor de ponderación en los casos en los que la predicción residual inter-vista está habilitada y aplicada (338).

El decodificador de vídeo 30 puede entonces determinar un bloque de referencia de disparidad en el mismo conjunto de acceso que el bloque actual (352). Por ejemplo, el decodificador de vídeo 30 puede determinar un vector de disparidad para ubicar un bloque de referencia de disparidad que tiene el mismo valor de POC que el bloque actual, pero ubicado en una segunda vista diferente. En algunos casos, el decodificador de vídeo 30 puede determinar el vector de disparidad en función de los datos incluidos en el flujo de bits. En otros casos, el decodificador de vídeo 30 puede aplicar el mismo procedimiento que el codificador de vídeo 20 para determinar el vector de disparidad.

El decodificador de vídeo 30 puede determinar si un búfer de imágenes decodificadas (también denominado memoria de imágenes de referencia en esta invención) contiene una imagen que tiene un valor de POC que es igual al valor de POC de la imagen de referencia temporal (354). Por ejemplo, el decodificador de vídeo 30 puede determinar si la imagen indicada por una combinación del vector de movimiento temporal y el vector de movimiento de disparidad está incluida en el búfer de imágenes decodificadas. En algunos casos, incluso si la imagen de referencia de disparidad temporal potencial se incluye en el búfer de imágenes decodificadas, el decodificador de vídeo 30 puede determinar además si la imagen se incluye en una o más listas de imágenes de referencia para el bloque de referencia de disparidad.

Si la imagen de referencia de disparidad temporal potencial se incluye en el búfer de imágenes decodificadas (y/o listas de imágenes de referencia del bloque de referencia de disparidad), el decodificador de vídeo 30 puede realizar un procedimiento de predicción residual inter-vista para predecir los datos residuales del bloque actual (356). Por ejemplo, como se señaló anteriormente, el decodificador de vídeo 30 puede determinar un bloque de referencia de disparidad indicado por un vector de disparidad del primer bloque, determinar un bloque de referencia de disparidad temporal aplicando el vector de movimiento del bloque actual al bloque de referencia de disparidad y determinar un predictor residual en función de una diferencia entre el bloque de referencia de disparidad temporal y el bloque de referencia de disparidad. El decodificador de vídeo 30 también puede aplicar un factor de ponderación, como se señala en el flujo de bits, al predictor residual.

Si la imagen de referencia de disparidad temporal potencial no está incluida en el búfer de imágenes decodificadas (y/o las listas de imágenes de referencia del bloque de referencia de disparidad) (la rama no de la etapa 354), el decodificador de vídeo 30 puede modificar el procedimiento de predicción residual inter-vista (358). En algunos ejemplos, el decodificador de vídeo 30 puede modificar el procedimiento al deshabilitar el procedimiento. En otros ejemplos, el decodificador de vídeo 30 puede seleccionar una imagen de referencia disponible (una imagen de referencia que se incluye en el búfer de imágenes decodificadas y/o la lista de imágenes de referencia) y escalar el vector de movimiento temporal en consecuencia.

En cualquier caso, el decodificador de vídeo 30 puede predecir el bloque actual utilizando el vector de movimiento decodificado (360). El decodificador de vídeo 30 puede entonces escanear inversamente los coeficientes reproducidos (362), para crear un bloque de coeficientes de transformación cuantificados. El decodificador de vídeo 30 puede entonces cuantificar inversamente y transformar inversamente los coeficientes para producir un bloque residual (364). El decodificador de vídeo 30 puede decodificar en última instancia el bloque actual combinando el bloque predicho y el o los bloques residuales (366). Por ejemplo, en casos en los que no se aplica la predicción residual inter-vista, el decodificador de vídeo 30 puede combinar simplemente el bloque predicho y el residual decodificado. En los casos en los que se aplica la predicción residual inter-vista, el decodificador de vídeo 30 puede combinar el bloque predicho, el residual decodificado (que representa un residual final) y el predictor residual.

Se debe reconocer que, dependiendo del ejemplo, determinados actos o eventos de cualquiera de las técnicas descritas en esta solicitud se pueden realizar en una secuencia diferente, se pueden añadir, fusionar u omitir todos

juntos (por ejemplo, no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Es más, en determinados ejemplos, los actos o eventos se pueden realizar simultáneamente, por ejemplo, a través de procesamiento de multihilo, procesamiento interrumpido o múltiples procesadores, en lugar de secuencialmente.

5 Ciertos aspectos de esta descripción se han descrito con respecto al estándar HEVC en desarrollo con fines ilustrativos. Sin embargo, las técnicas descritas en esta divulgación pueden ser útiles para otros procedimientos de codificación de vídeo, incluidos otros procedimientos de codificación de vídeo estándar o patentados aún no desarrollados.

10 Las técnicas descritas anteriormente pueden realizarse mediante el codificador de vídeo 20 (Figuras 1 y 2) y/o el decodificador de vídeo 30 (Figuras 1 y 3), ambos de los cuales pueden denominarse generalmente codificador de vídeo. Del mismo modo, la codificación de vídeo puede referirse a la codificación de vídeo o decodificación de vídeo, según corresponda.

15 Debe entenderse que, dependiendo del ejemplo, ciertos actos o eventos de cualquiera de los procedimientos descritos en esta invención se pueden realizar en una secuencia diferente, se pueden agregar, fusionar o omitir por completo (por ejemplo, no todos los actos o eventos descritos). eventos son necesarios para la práctica del procedimiento). Es más, en determinados ejemplos, los actos o eventos se pueden realizar simultáneamente, por ejemplo, a través de procesamiento de multihilo, procesamiento interrumpido o múltiples procesadores, en lugar de secuencialmente.

20 Además, si bien ciertos aspectos de esta divulgación se describen como realizados por un único módulo o conjunto con fines de claridad, debe entenderse que las técnicas de esta divulgación pueden realizarse mediante una combinación de conjuntos o módulos asociados con un codificador de vídeo.

25 Si bien se describieron anteriormente combinaciones particulares de diversos aspectos de las técnicas, estas combinaciones se proporcionan meramente para ilustrar ejemplos de las técnicas descritas en esta invención divulgación. Por consiguiente, las técnicas de esta divulgación no deben limitarse a estas combinaciones de ejemplo y pueden abarcar cualquier combinación concebible de los diversos aspectos de las técnicas descritas en esta divulgación.

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

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

40 A modo de ejemplo, y sin limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda utilizar para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión se denomina correctamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota utilizando un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL) o tecnologías inalámbricas como infrarrojos, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o tecnologías inalámbricas como infrarrojos, radio y microondas se incluyen en la definición de medio.

45 Debería entenderse, sin embargo, 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 están dirigidos a medios de almacenamiento tangibles no transitorios. Disco, como se usa en esta invención, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco Blu-ray, donde unos discos generalmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Dentro del alcance de los medios legibles por ordenador también deben incluirse combinaciones de lo anterior.

50 Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables en campo (FP-GA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", como se usa en esta invención, puede referirse a cualquiera de las estructuras

anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en esta invención. Además, en algunos aspectos, la funcionalidad descrita en esta invención puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

5 Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen varios componentes, módulos o conjuntos para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas descritas, pero no necesariamente requieren su realización por
10 diferentes conjuntos de hardware. Más bien, como se ha descrito anteriormente, varios conjuntos pueden combinarse en un conjunto de hardware de códec o proporcionarse por una colección de conjuntos de hardware interoperativos, que incluyen uno o más procesadores como se ha descrito anteriormente, junto con software y/o firmware adecuados.

15 Diversos aspectos de la descripción se han descrito anteriormente. Estos y otros aspectos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para decodificar datos de vídeo multicapa, comprendiendo el procedimiento:

determinar, para un primer conjunto de codificación de datos de vídeo en una primera ubicación temporal en una primera vista, si una o más listas de imágenes de referencia para codificar el primer conjunto de codificación contienen al menos una imagen de referencia temporal asociada con la misma vista que el primer conjunto de codificación, en una segunda ubicación temporal diferente;
en base a la determinación de que la una o más listas de imágenes de referencia para codificar el primer conjunto de codificación no contienen al menos una imagen de referencia temporal, deshabilitar un procedimiento de predicción residual inter-vista, donde deshabilitar el procedimiento de predicción residual inter-vista comprende omitir la decodificación de un factor de ponderación para el procedimiento de predicción residual inter-vista para el primer conjunto de codificación y cada otro conjunto de codificación de una imagen que contiene el primer conjunto de codificación, y donde el procedimiento de predicción residual inter-vista incluye:

determinar, para el primer conjunto de codificación, un vector de movimiento temporal que indica una imagen de referencia temporal;
determinar un bloque de referencia de disparidad en una imagen de referencia de disparidad en la primera ubicación temporal en una segunda vista que es diferente de la primera vista, donde el bloque de referencia de disparidad se indica mediante un vector de disparidad asociado con el primer conjunto de codificación;
determinar un bloque de referencia de disparidad temporal en la segunda ubicación temporal en la segunda vista, donde el bloque de referencia de disparidad temporal se indica mediante el vector de movimiento temporal y el vector de disparidad en combinación; y
determinar un predictor residual para predecir datos residuales asociados con el primer conjunto de codificación, donde el predictor residual se determina en función de las diferencias entre las muestras del bloque de referencia de disparidad y las muestras correspondientes del bloque de referencia de disparidad temporal, y donde los datos residuales asociados con el primer conjunto de codificación representan diferencias entre las muestras del primer conjunto de codificación y las muestras correspondientes de un bloque de referencia en la imagen de referencia temporal; y

decodificar el primer conjunto de codificación de datos de vídeo con respecto a al menos un bloque de referencia de datos de vídeo de una imagen de referencia en las una o más listas de imágenes de referencia.

2. Un procedimiento para codificar datos de vídeo multicapa, comprendiendo el procedimiento:

determinar, para un primer conjunto de codificación de datos de vídeo en una primera ubicación temporal en una primera vista, si una o más listas de imágenes de referencia para codificar el primer conjunto de codificación contienen al menos una imagen de referencia temporal asociada con la misma vista que el primer conjunto de codificación, en una segunda ubicación temporal diferente;
en base a la determinación de que la una o más listas de imágenes de referencia para codificar el primer conjunto de codificación no contienen al menos una imagen de referencia temporal, deshabilitar un procedimiento de predicción residual inter-vista, donde deshabilitar el procedimiento de predicción residual inter-vista comprende omitir la codificación de un factor de ponderación para el procedimiento de predicción residual inter-vista para el primer conjunto de codificación y cada otro conjunto de codificación de una imagen que contiene el primer conjunto de codificación, y donde el procedimiento de predicción residual inter-vista incluye:

determinar, para el primer conjunto de codificación, un vector de movimiento temporal que indica una imagen de referencia temporal;
determinar un bloque de referencia de disparidad en una imagen de referencia de disparidad en la primera ubicación temporal en una segunda vista que es diferente de la primera vista, donde el bloque de referencia de disparidad se indica mediante un vector de disparidad asociado con el primer conjunto de codificación;
determinar un bloque de referencia de disparidad temporal en la segunda ubicación temporal en la segunda vista, donde el bloque de referencia de disparidad temporal se indica mediante el vector de movimiento temporal y el vector de disparidad en combinación; y
determinar un predictor residual para predecir datos residuales asociados con el primer conjunto de codificación, donde el predictor residual se determina en función de las diferencias entre las muestras del bloque de referencia de disparidad y las muestras correspondientes del bloque de referencia de disparidad temporal, y donde los datos residuales asociados con el primer conjunto de codificación representan diferencias entre las muestras del primer conjunto de codificación y las muestras correspondientes de un bloque de referencia en la imagen de referencia temporal; y

codificar el primer conjunto de codificación de datos de vídeo con respecto a al menos un bloque de referencia de datos de vídeo de una imagen de referencia en las una o más listas de imágenes de referencia.

3. El procedimiento según la reivindicación 1 o 2, donde determinar si la una o más listas de imágenes de

referencia incluyen al menos una imagen de referencia que comprende determinar si una imagen que contiene el primer bloque es una imagen de acceso aleatorio.

4. El procedimiento según la reivindicación 1 o 2, que comprende además determinar automáticamente que el factor de ponderación es cero cuando se omite la codificación del factor de ponderación.

5. El procedimiento según la reivindicación 1 o 2, que comprende además:
basado en la determinación de que la una o más listas de imágenes de referencia para codificar el primer conjunto de codificación contienen al menos una imagen de referencia en la segunda ubicación temporal, no deshabilitar el procedimiento de predicción residual inter-vista.

6. El procedimiento según la reivindicación 1 o 2, donde la una o más listas de imágenes de referencia comprenden una primera lista de imágenes de referencia y una segunda lista de imágenes de referencia, donde determinar si la una o más listas de imágenes de referencia contienen la al menos una imagen de referencia en la segunda ubicación temporal comprende determinar si cualquiera de la primera lista de imágenes de referencia y la segunda lista de imágenes de referencia contiene la al menos una imagen de referencia en la segunda ubicación temporal.

7. Un aparato para decodificar datos de vídeo multi-capa, comprendiendo el aparato:

medios para determinar, para un primer conjunto de codificación de datos de vídeo en una primera ubicación temporal, si una o más listas de imágenes de referencia para codificar el primer conjunto de codificación contienen al menos una imagen de referencia temporal asociada con la misma vista que el primer conjunto de codificación, en una segunda ubicación temporal diferente;

medios para deshabilitar un procedimiento de predicción residual inter-vista, basándose en la determinación de que la una o más listas de imágenes de referencia para codificar el primer conjunto de codificación no contienen al menos una imagen de referencia temporal, donde deshabilitar el procedimiento de predicción residual inter-vista comprende omitir la decodificación de un factor de ponderación para el procedimiento de predicción residual inter-vista para el primer conjunto de codificación y cada otro conjunto de codificación de una imagen que contiene el primer conjunto de codificación, y donde el procedimiento de predicción residual inter-vista incluye:

determinar, para el primer conjunto de codificación, un vector de movimiento temporal que indica una imagen de referencia temporal;

determinar un bloque de referencia de disparidad en una imagen de referencia de disparidad en la primera ubicación temporal en una segunda vista que es diferente de la primera vista, donde el bloque de referencia de disparidad se indica mediante un vector de disparidad asociado con el primer conjunto de codificación;

determinar un bloque de referencia de disparidad temporal en la segunda ubicación temporal en la segunda vista, donde el bloque de referencia de disparidad temporal se indica mediante el vector de movimiento temporal y el vector de disparidad en combinación; y

determinar un predictor residual para predecir datos residuales asociados con el primer conjunto de codificación, donde el predictor residual se determina en función de las diferencias entre las muestras del bloque de referencia de disparidad y las muestras correspondientes del bloque de referencia de disparidad temporal, y donde los datos residuales asociados con el primer conjunto de codificación representan diferencias entre las muestras del primer conjunto de codificación y las muestras correspondientes de un bloque de referencia en la imagen de referencia temporal; y

medios para decodificar el primer conjunto de codificación de datos de vídeo con respecto a al menos un bloque de referencia de datos de vídeo de una imagen de referencia en las una o más listas de imágenes de referencia.

8. Un aparato para codificar datos de vídeo de múltiples capas, comprendiendo el aparato:

medios para determinar, para un primer conjunto de codificación de datos de vídeo en una primera ubicación temporal, si una o más listas de imágenes de referencia para codificar el primer conjunto de codificación contienen al menos una imagen de referencia temporal asociada con la misma vista que el primer conjunto de codificación, en una segunda ubicación temporal diferente;

medios para deshabilitar un procedimiento de predicción residual inter-vista, basándose en la determinación de que la una o más listas de imágenes de referencia para codificar el primer conjunto de codificación no contienen al menos una imagen de referencia temporal, donde deshabilitar el procedimiento de predicción residual inter-vista comprende omitir la codificación de un factor de ponderación para el procedimiento de predicción residual inter-vista para el primer conjunto de codificación y cada otro conjunto de codificación de una imagen que contiene el primer conjunto de codificación, y donde el procedimiento de predicción residual inter-vista incluye:

determinar, para el primer conjunto de codificación, un vector de movimiento temporal que indica la imagen de referencia temporal;

determinar un bloque de referencia de disparidad en una imagen de referencia de disparidad en la primera

ubicación temporal en una segunda vista que es diferente de la primera vista, donde el bloque de referencia de disparidad se indica mediante un vector de disparidad asociado con el primer conjunto de codificación; determinar un bloque de referencia de disparidad temporal en la segunda ubicación temporal en la segunda vista, donde el bloque de referencia de disparidad temporal se indica mediante el vector de movimiento temporal y el vector de disparidad en combinación; y
 5 determinar un predictor residual para predecir datos residuales asociados con el primer conjunto de codificación, donde el predictor residual se determina en función de las diferencias entre las muestras del bloque de referencia de disparidad y las muestras correspondientes del bloque de referencia de disparidad temporal, y donde los datos residuales asociados con el primer conjunto de codificación representan diferencias
 10 entre las muestras del primer conjunto de codificación y las muestras correspondientes de un bloque de referencia en la imagen de referencia temporal; y

medios para codificar el primer conjunto de codificación de datos de vídeo con respecto a al menos un bloque de referencia de datos de vídeo de una imagen de referencia en la una o más listas de imágenes de referencia.

9. El aparato según la reivindicación 7, que comprende además:

medios para, basándose en la determinación de que la una o más listas de imágenes de referencia para codificar el primer conjunto de codificación contienen al menos una imagen de referencia en la segunda ubicación temporal, no deshabilitar el procedimiento de predicción residual inter-vista; y
 20 donde los medios para decodificar el primer conjunto de codificación con respecto, al menos, a un bloque de referencia comprenden medios para codificar los datos residuales para el primer conjunto de codificación con el procedimiento de predicción residual inter-vista que comprende:

medios para determinar el bloque de referencia temporal indicado por un vector de movimiento temporal del primer conjunto de codificación;
 medios para determinar el bloque de referencia de disparidad indicado por un vector de disparidad del primer conjunto de codificación;
 medios para determinar el bloque de referencia de disparidad temporal indicado por una combinación del vector de movimiento temporal y el vector de disparidad; y
 30 medios para decodificar los datos residuales para el primer conjunto de codificación usando el bloque de referencia temporal, el bloque de referencia de disparidad y el bloque de referencia de disparidad temporal.

10. El aparato según la reivindicación 8, que comprende además:

medios para, basándose en la determinación de que la una o más listas de imágenes de referencia para codificar el primer conjunto de codificación contienen al menos una imagen de referencia en la segunda ubicación temporal, no deshabilitar el procedimiento de predicción residual inter-vista; y
 40 donde los medios para codificar el primer conjunto de codificación con respecto a al menos un bloque de referencia comprenden medios para codificar los datos residuales para el primer conjunto de codificación con el procedimiento de predicción residual inter-vista que comprende:

medios para determinar el bloque de referencia temporal indicado por un vector de movimiento temporal del primer conjunto de codificación;
 45 medios para determinar el bloque de referencia de disparidad indicado por un vector de disparidad del primer conjunto de codificación;
 medios para determinar el bloque de referencia de disparidad temporal indicado por una combinación del vector de movimiento temporal y el vector de disparidad; y
 medios para codificar los datos residuales para el primer conjunto de codificación usando el bloque de referencia temporal, el bloque de referencia de disparidad y el bloque de referencia de disparidad temporal.

11. El aparato según la reivindicación 9, donde los medios para decodificar los datos residuales para el primer conjunto de codificación comprenden:

medios para obtener, a partir de un flujo de bits codificado, datos indicativos de un residuo final para el primer conjunto de codificación;
 medios para determinar el predictor residual en función de una diferencia entre el bloque de referencia de disparidad y el bloque de referencia de disparidad temporal; y
 60 medios para reconstruir el primer conjunto de codificación en función de una combinación del residuo final, el predictor residual y el bloque de referencia temporal.

12. El aparato según la reivindicación 11, donde los medios para codificar los datos residuales para el primer conjunto de codificación comprenden:

medios para determinar un primer residuo que comprende una diferencia entre el primer conjunto de codificación

y un bloque de referencia temporal;
medios para determinar un predictor residual que comprende una diferencia entre el bloque de referencia de
disparidad y el bloque de referencia de disparidad temporal;
medios para determinar un residuo final en función de una diferencia entre el primer residuo y el predictor residual;
medios para codificar datos que indiquen el residuo final en un flujo de bits.

13. Un medio no transitorio legible por ordenador que tiene instrucciones almacenadas en él que, cuando se
ejecutan, hacen que uno o más procesadores realicen el procedimiento según cualquiera de las reivindicaciones 1 a
6.

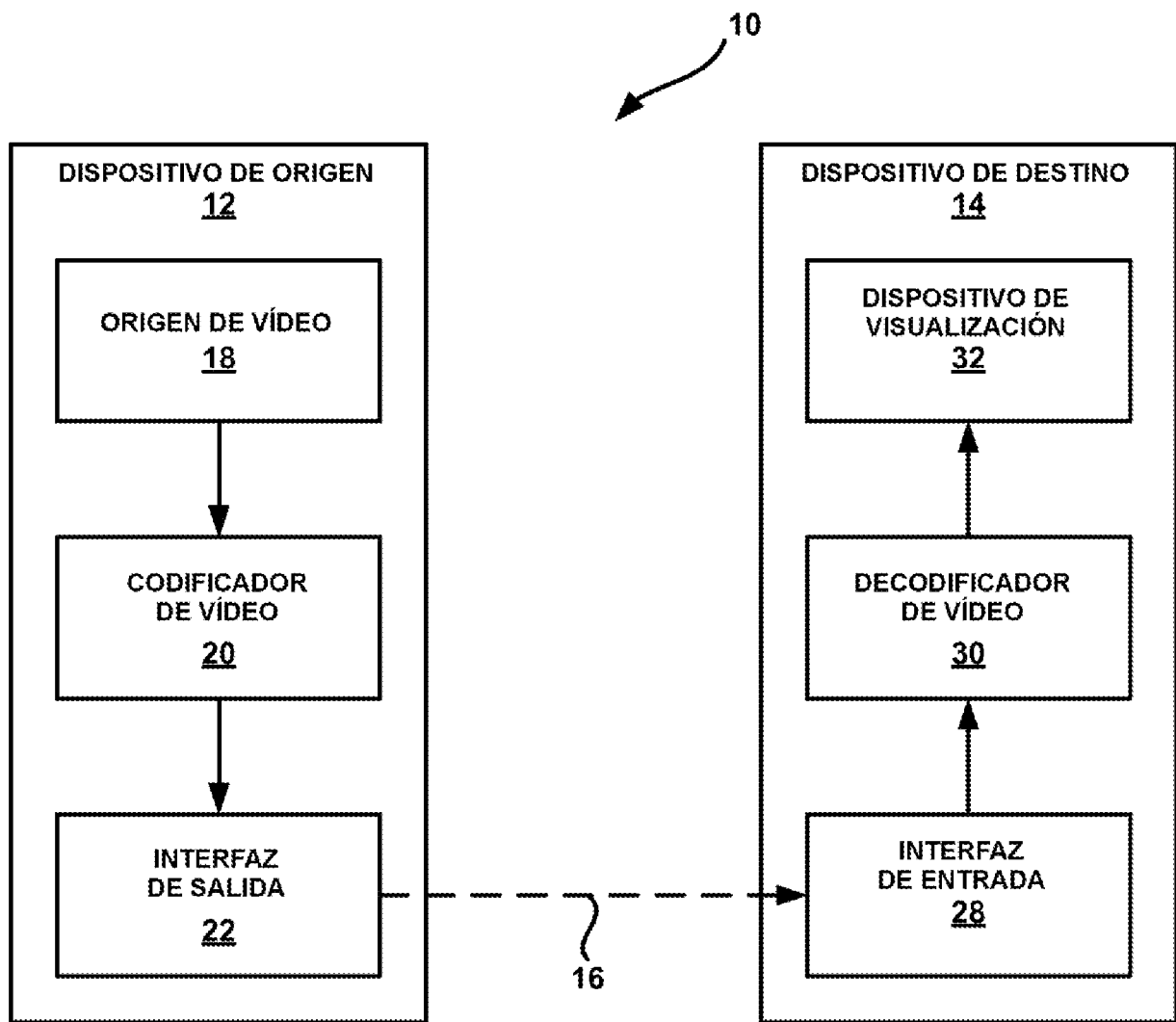
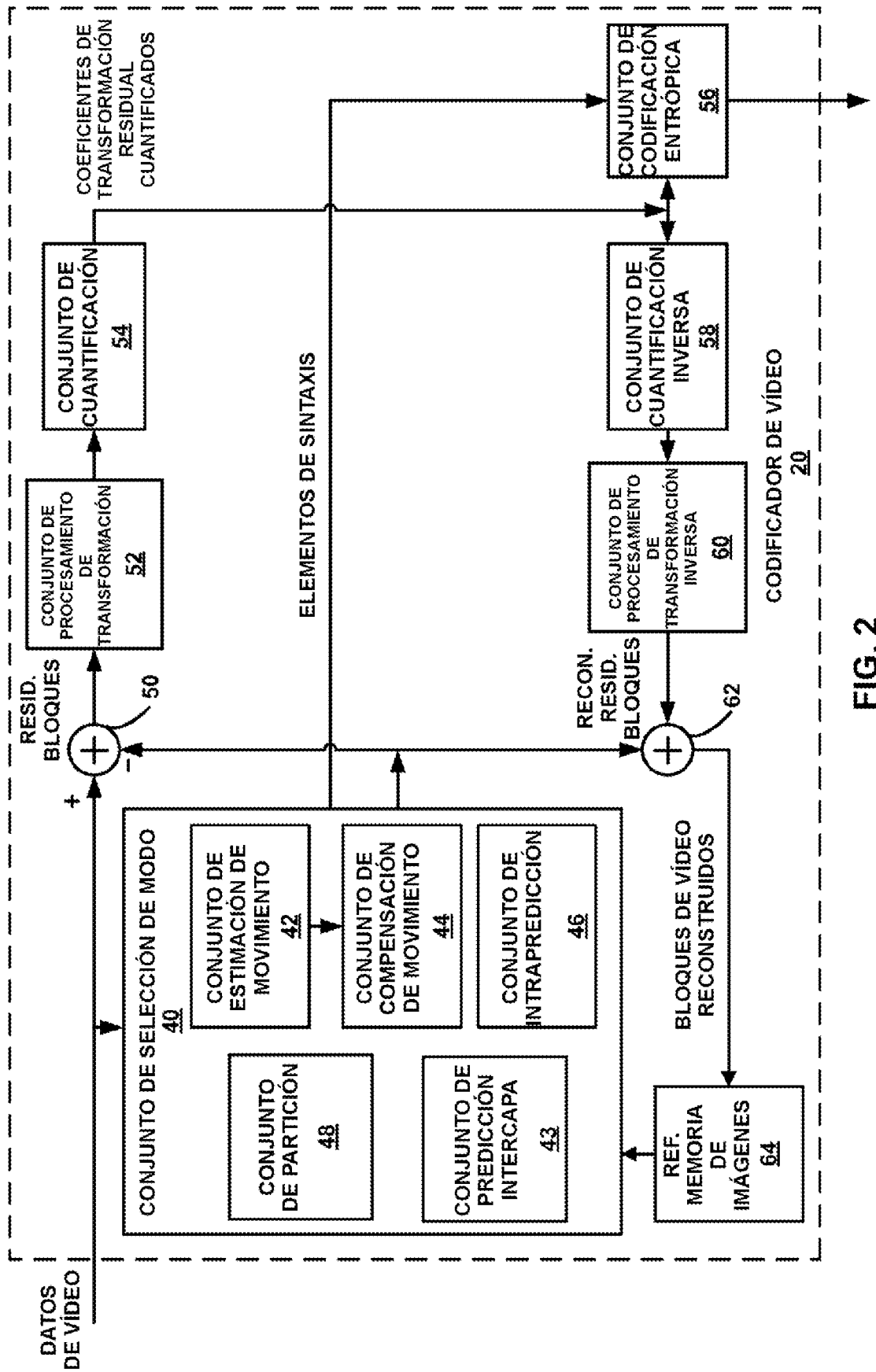


FIG. 1



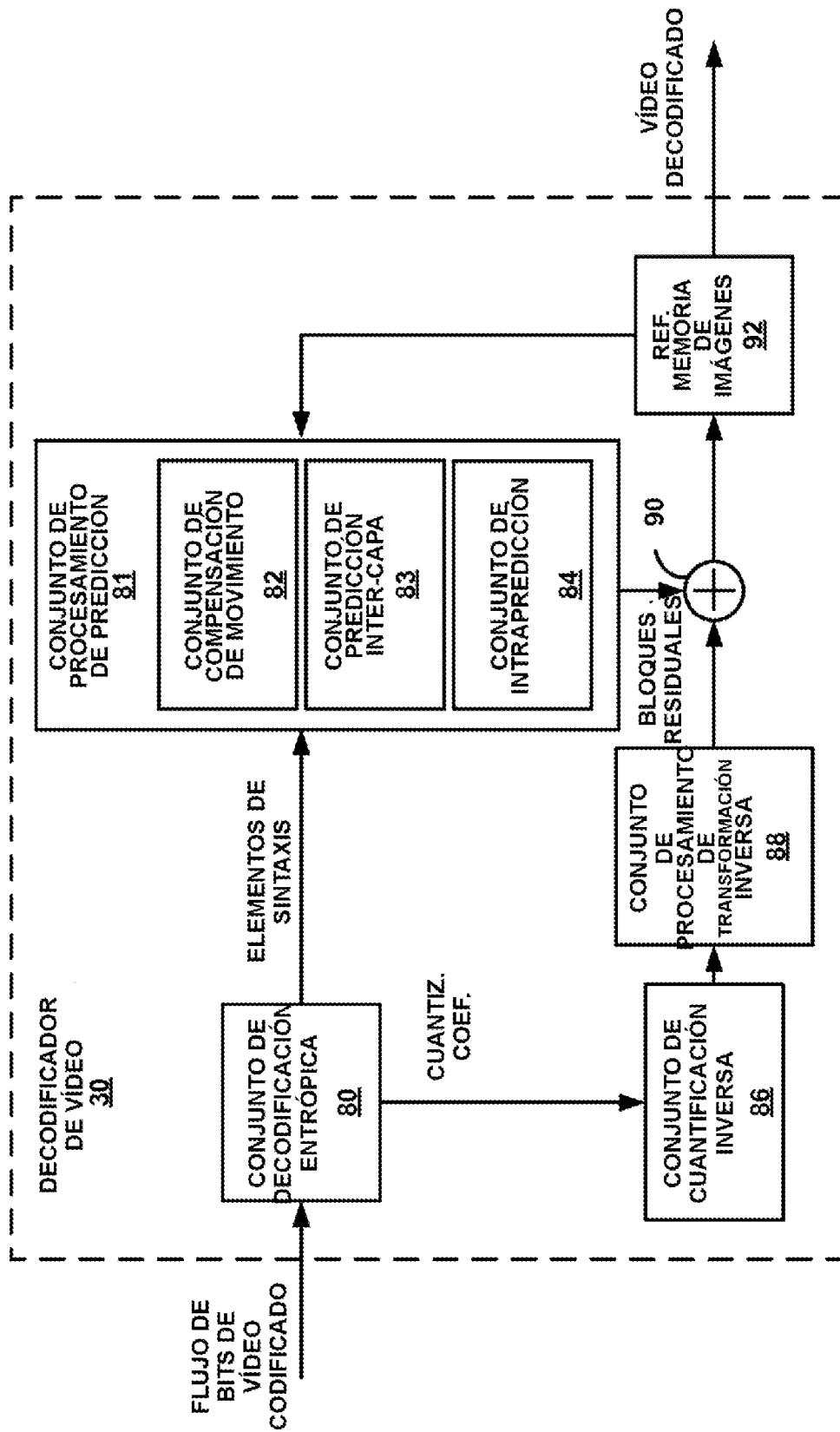


FIG. 3

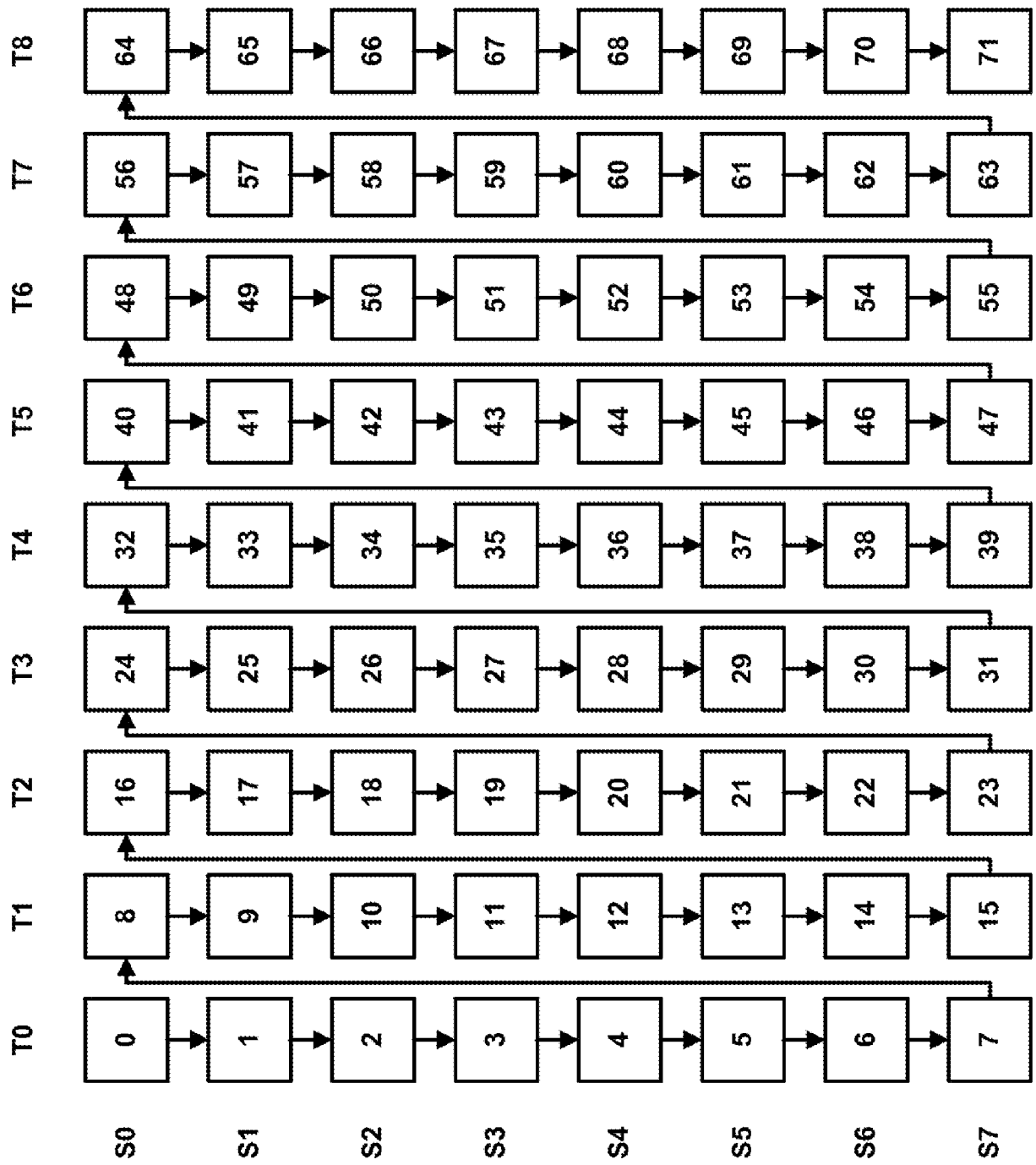


FIG. 4

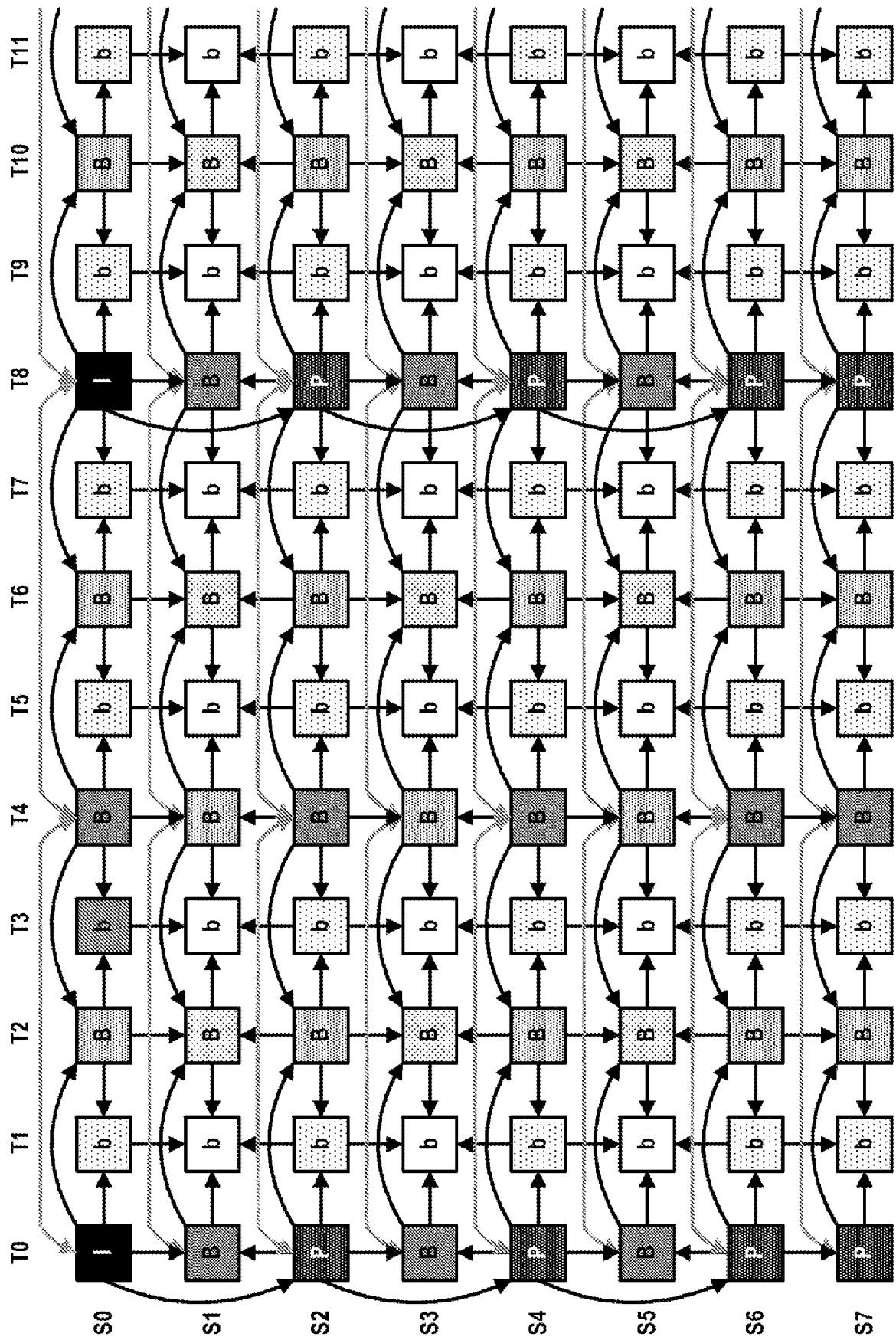


FIG. 5

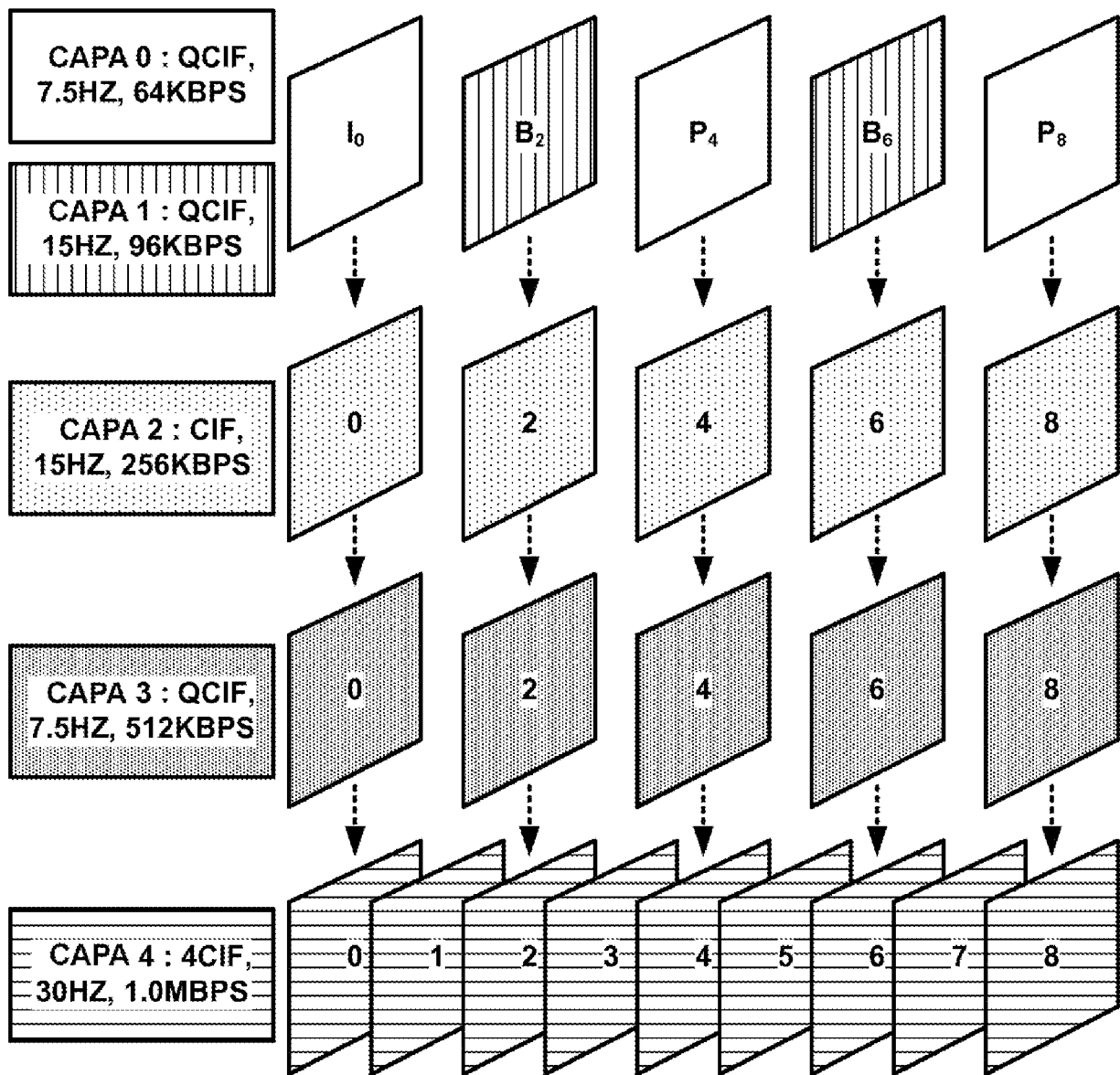


FIG. 6

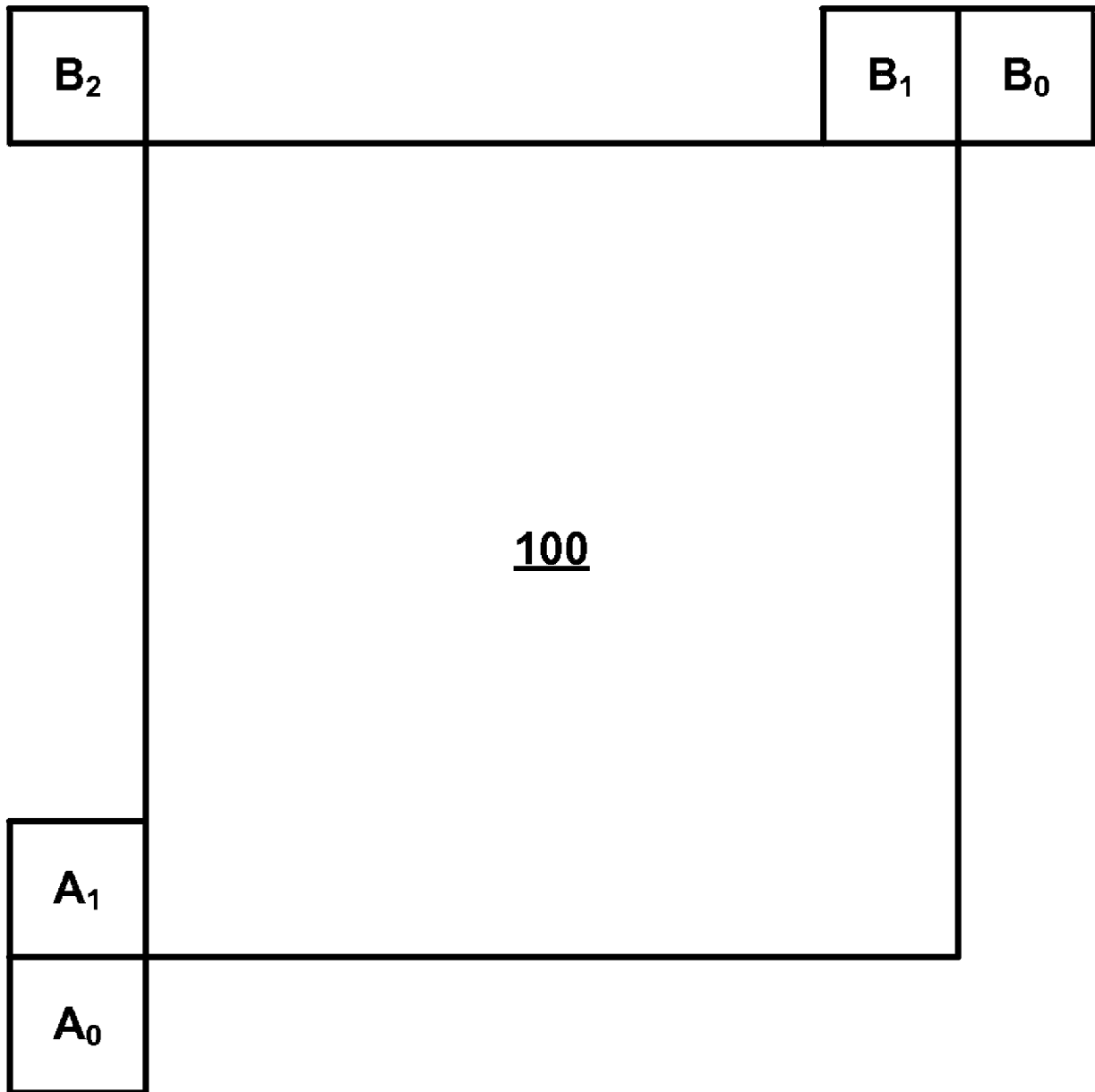


FIG. 7

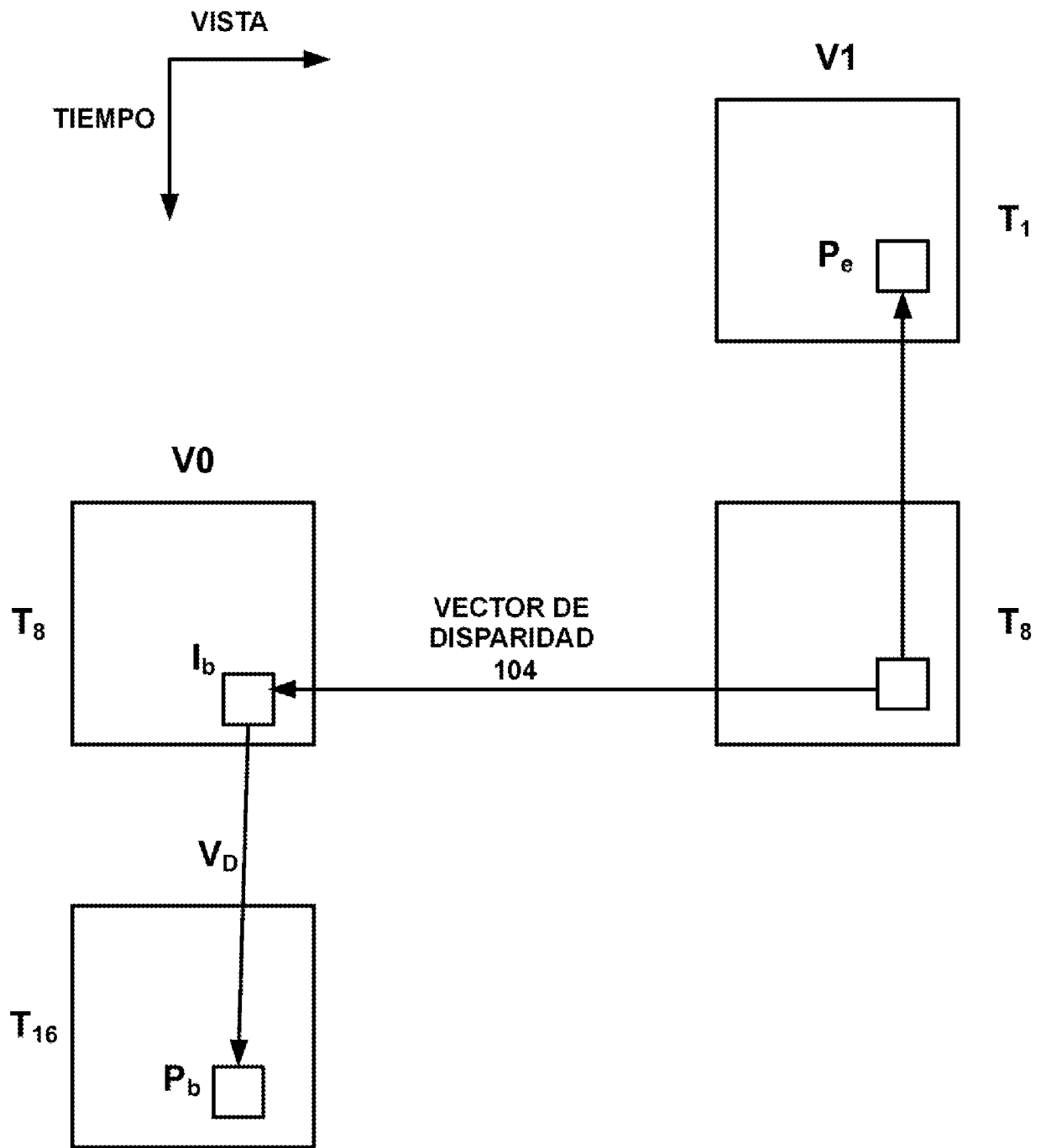


FIG. 8

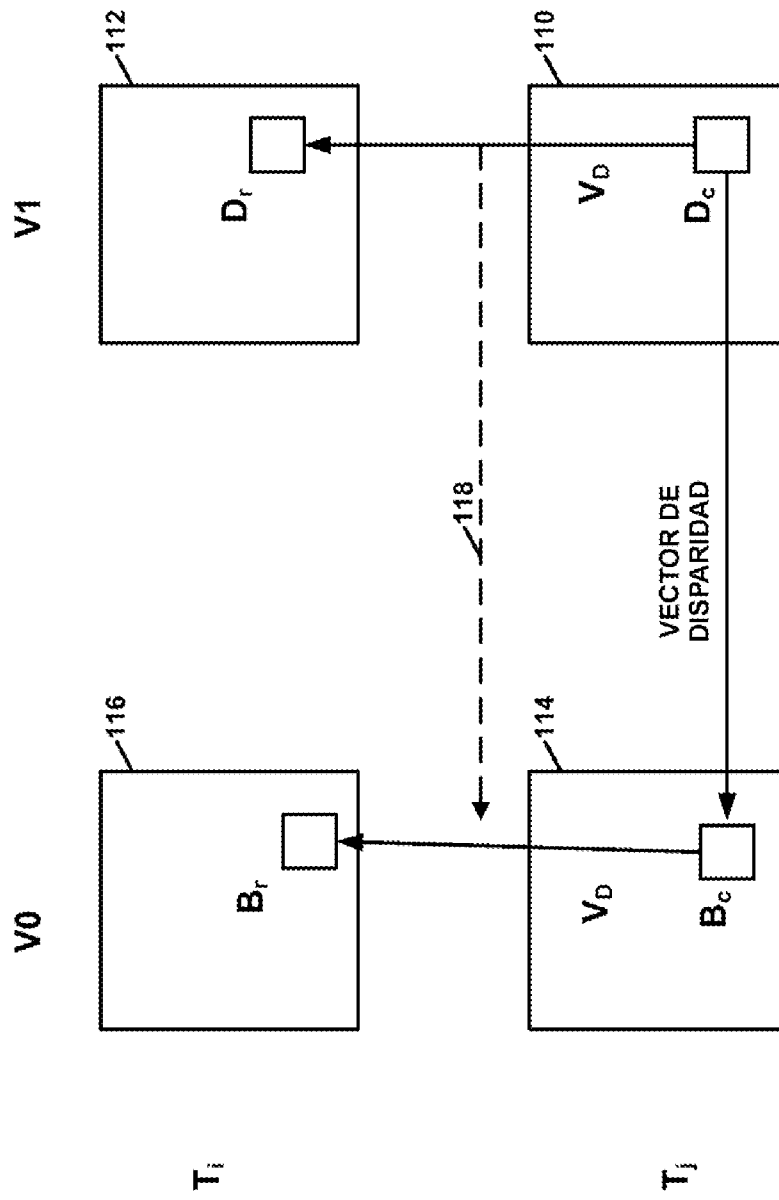


FIG. 9

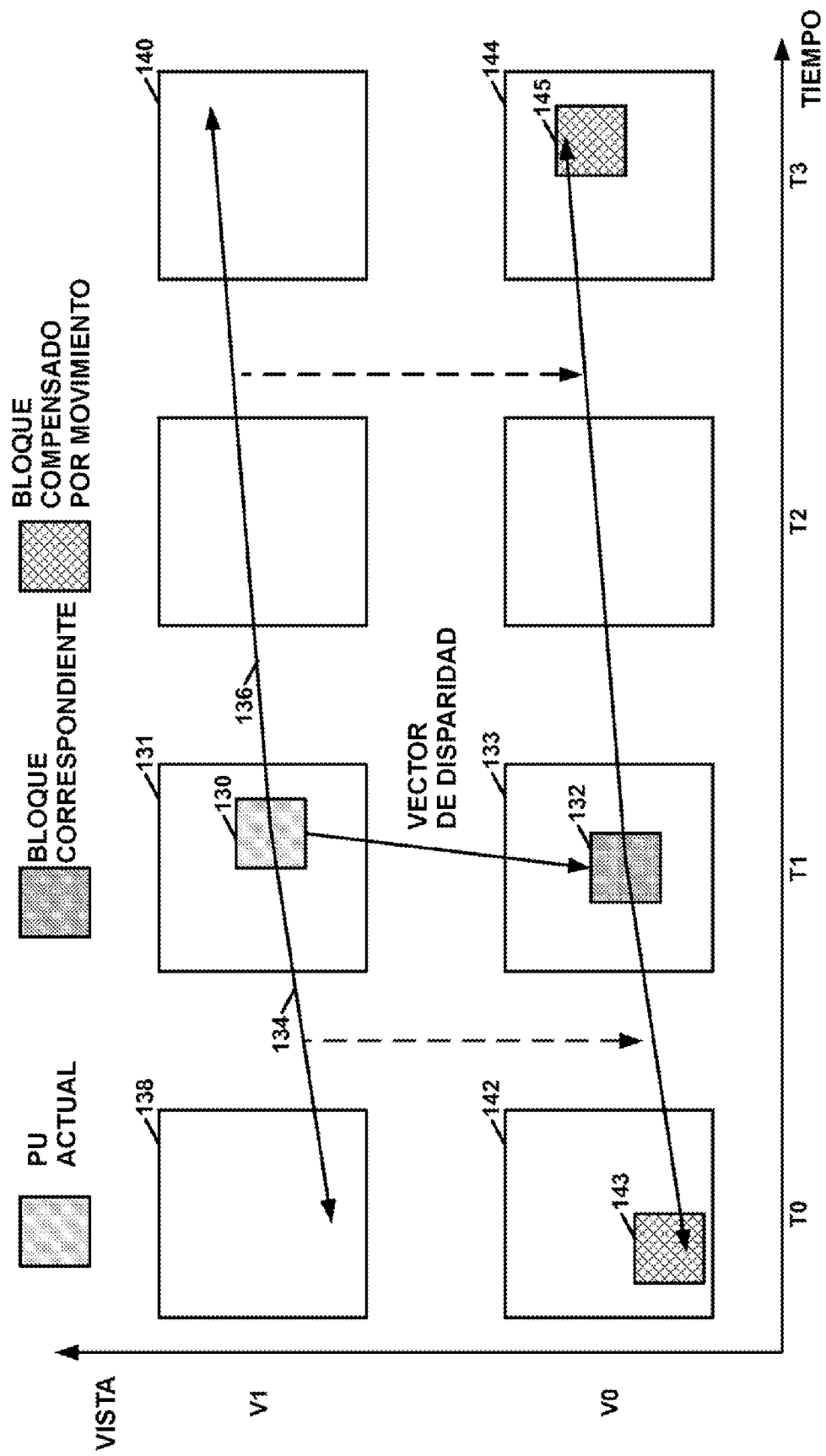


FIG. 10

A _{-1,-1}		A _{0,-1}	a _{0,-1}	b _{0,-1}	c _{0,-1}	A _{1,-1}		A _{2,-1}
A _{-1,0}		A _{0,0}	a _{0,0}	b _{0,0}	c _{0,0}	A _{1,0}		A _{2,0}
d _{-1,0}		d _{0,0}	e _{0,0}	f _{0,0}	g _{0,0}	d _{1,0}		d _{2,0}
h _{-1,0}		h _{0,0}	i _{0,0}	j _{0,0}	k _{0,0}	h _{1,0}		h _{2,0}
n _{-1,0}		n _{0,0}	p _{0,0}	q _{0,0}	r _{0,0}	n _{1,0}		n _{2,0}
A _{-1,1}		A _{0,1}	a _{0,1}	b _{0,1}	c _{0,1}	A _{1,1}		A _{2,1}
A _{-1,2}		A _{0,2}	a _{0,2}	b _{0,2}	c _{0,2}	A _{1,2}		A _{2,2}

FIG. 11

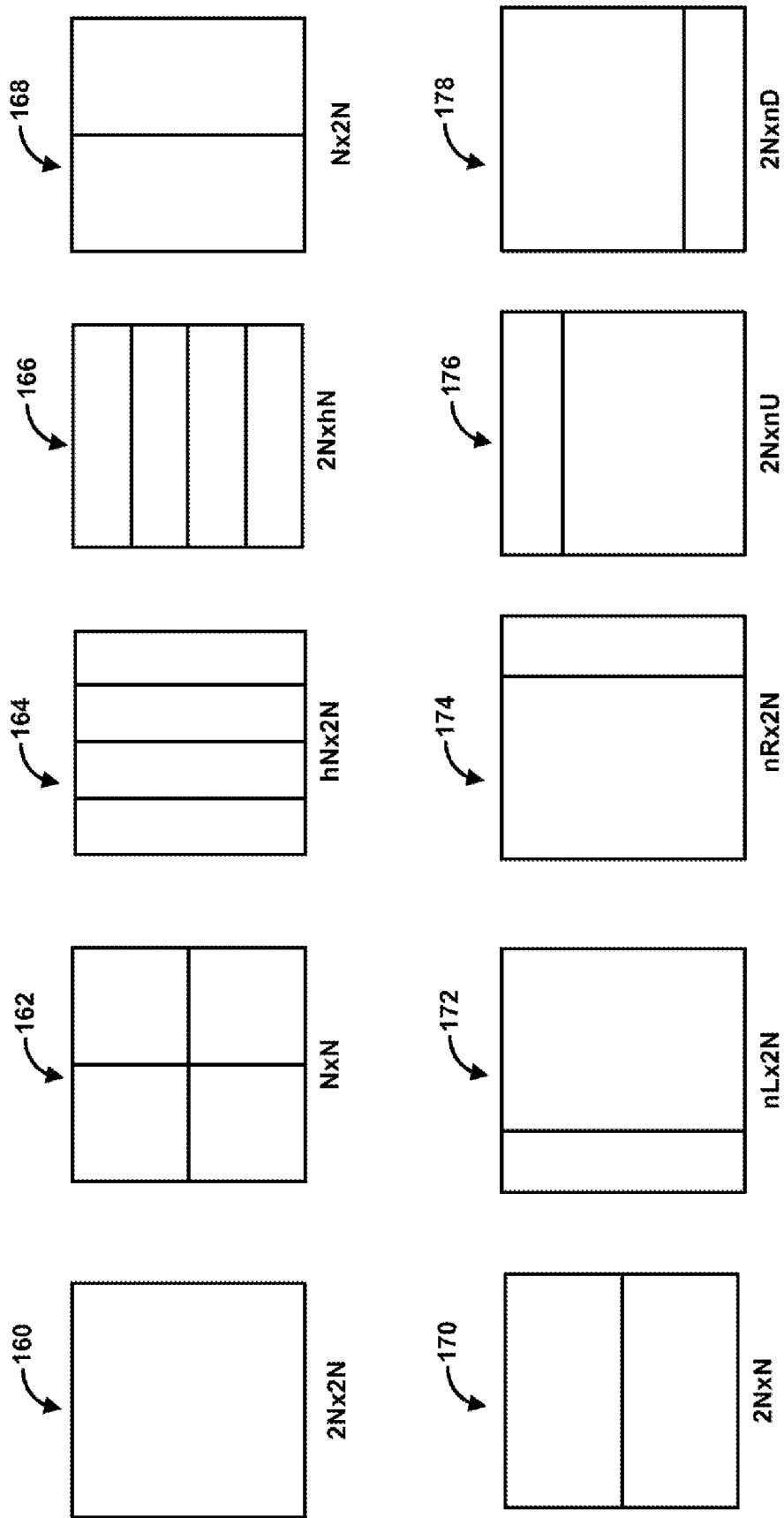


FIG. 12

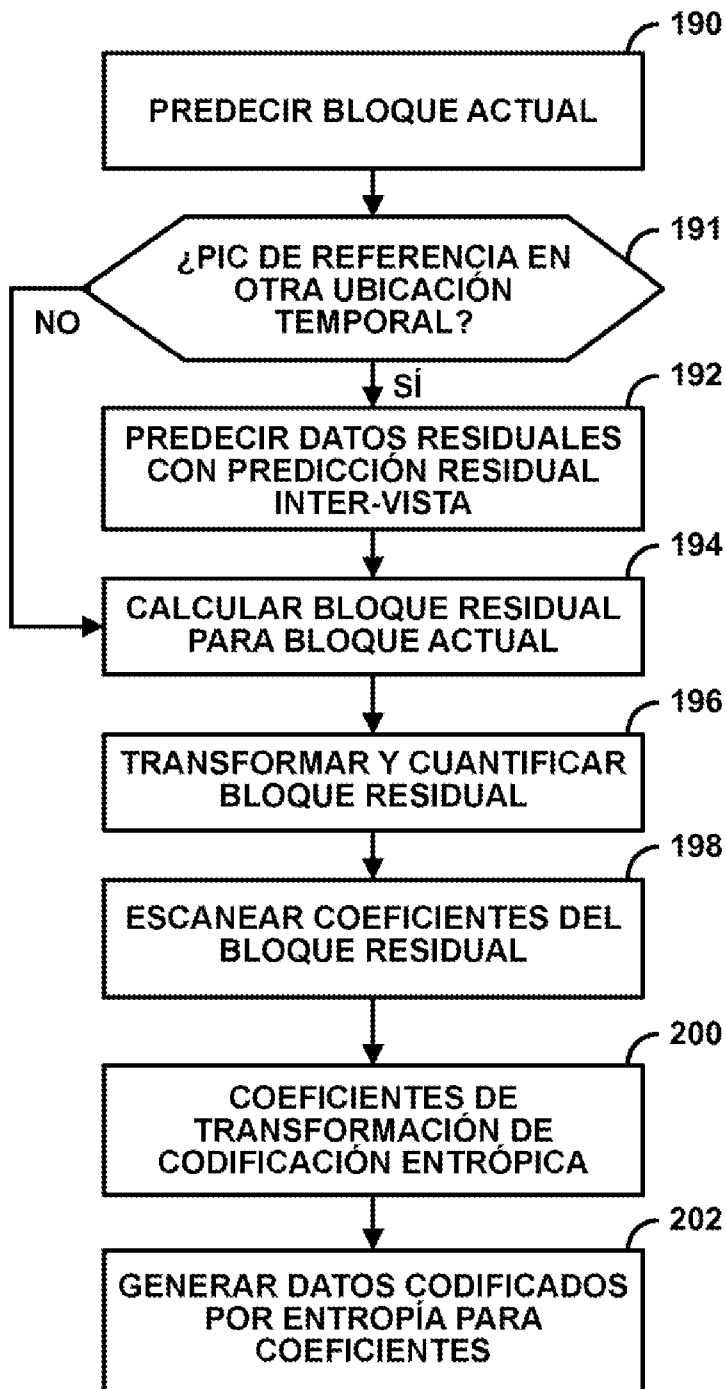


FIG. 13

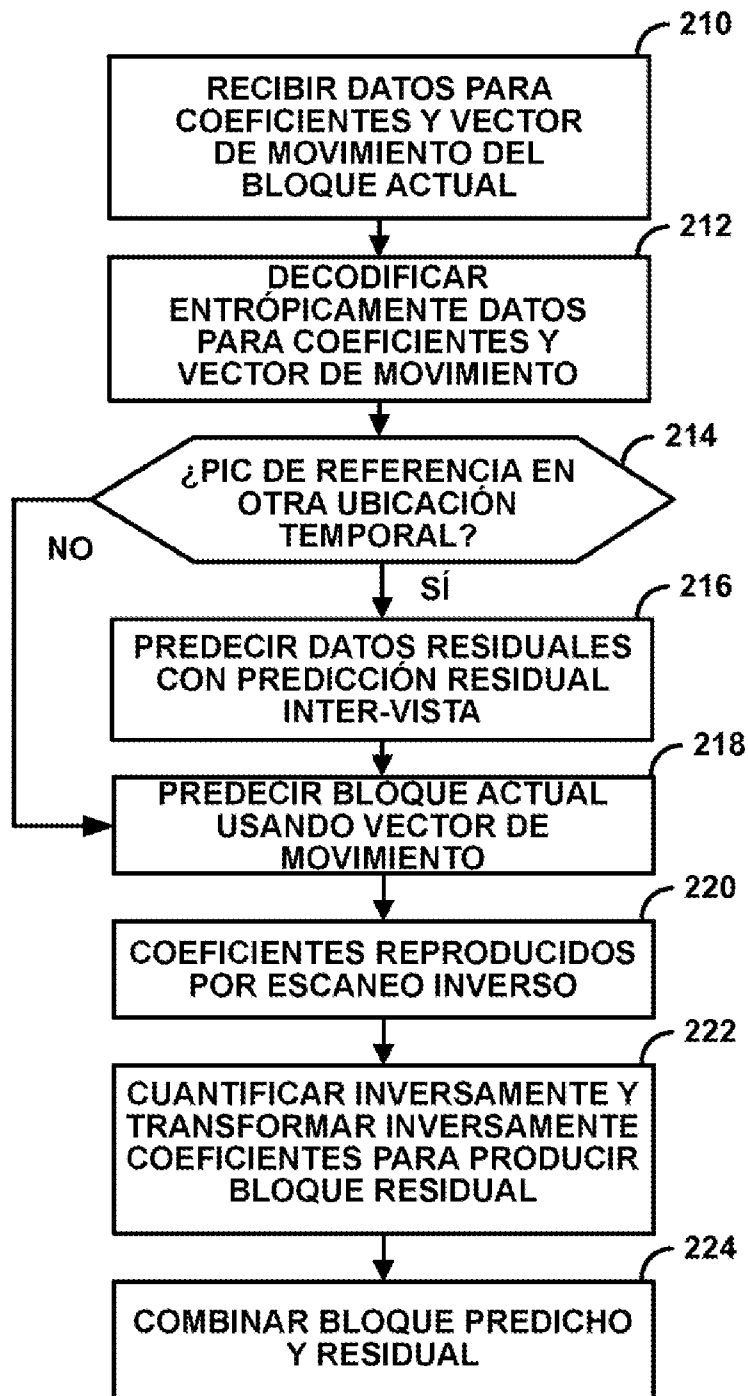


FIG. 14

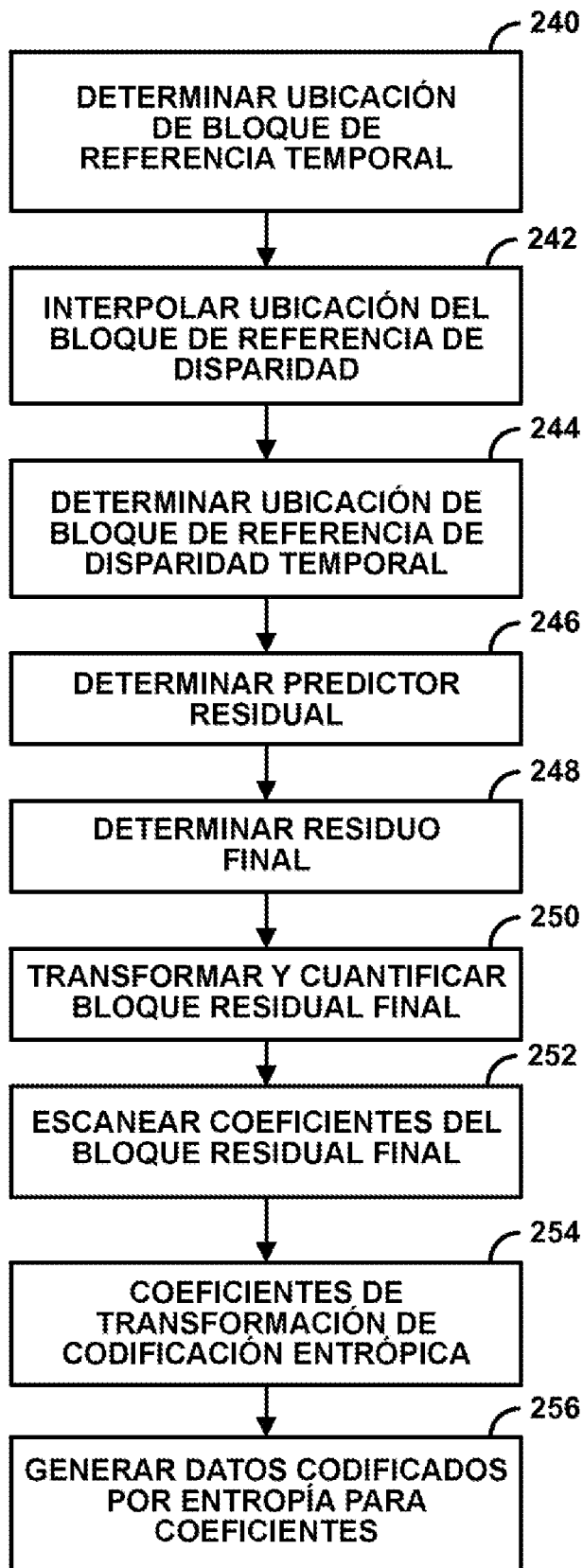


FIG. 15

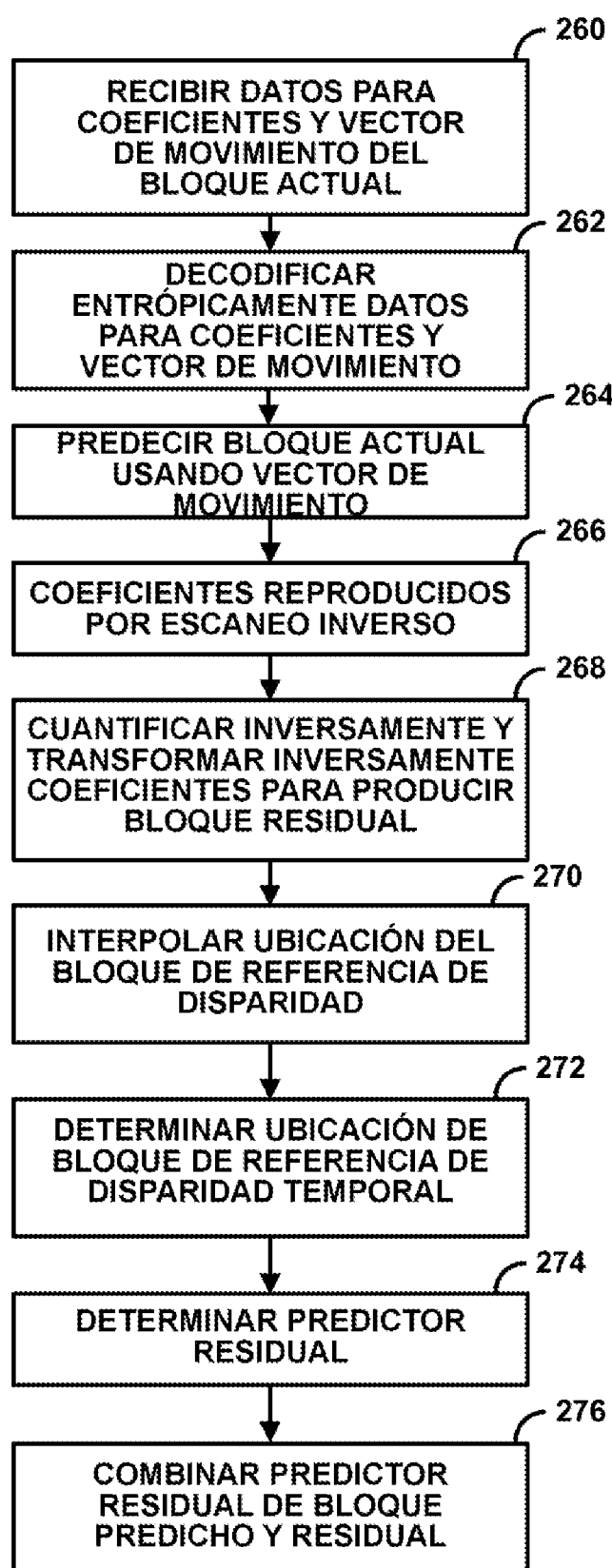


FIG. 16

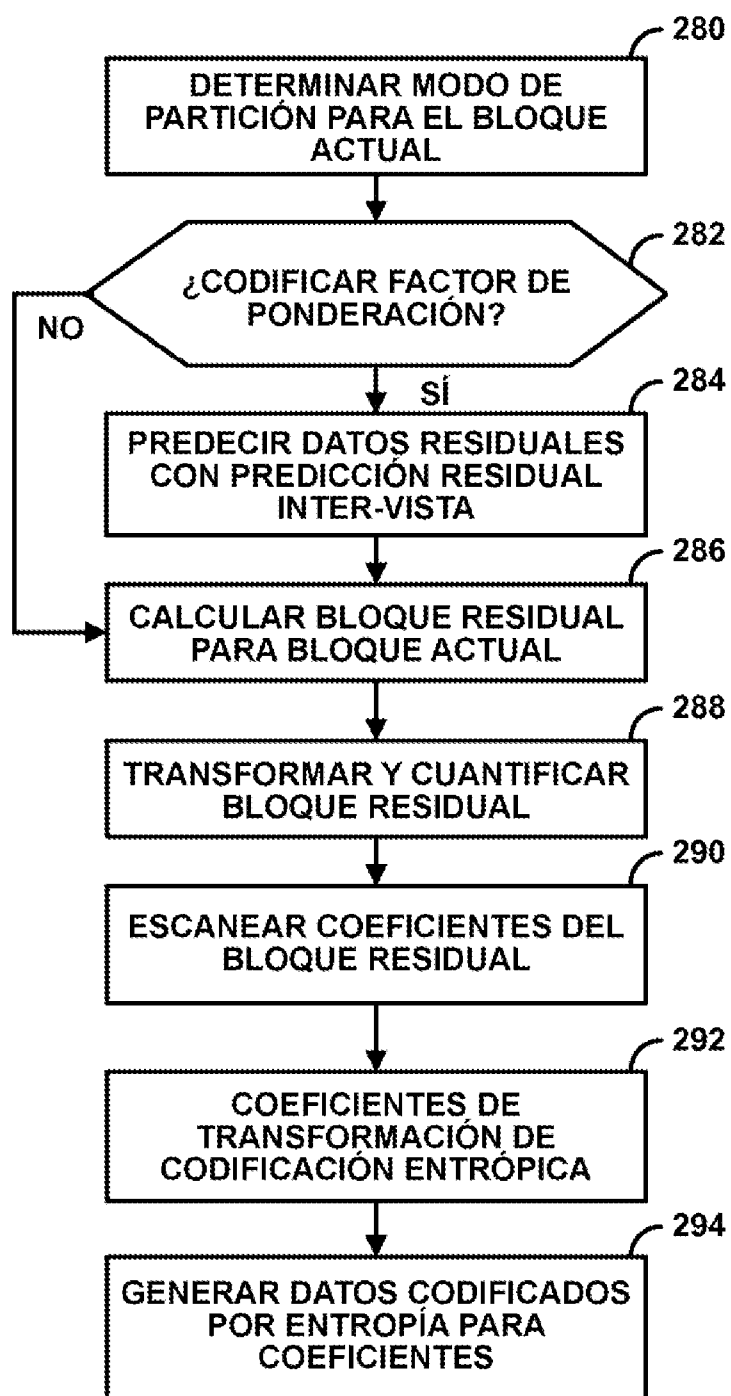


FIG. 17

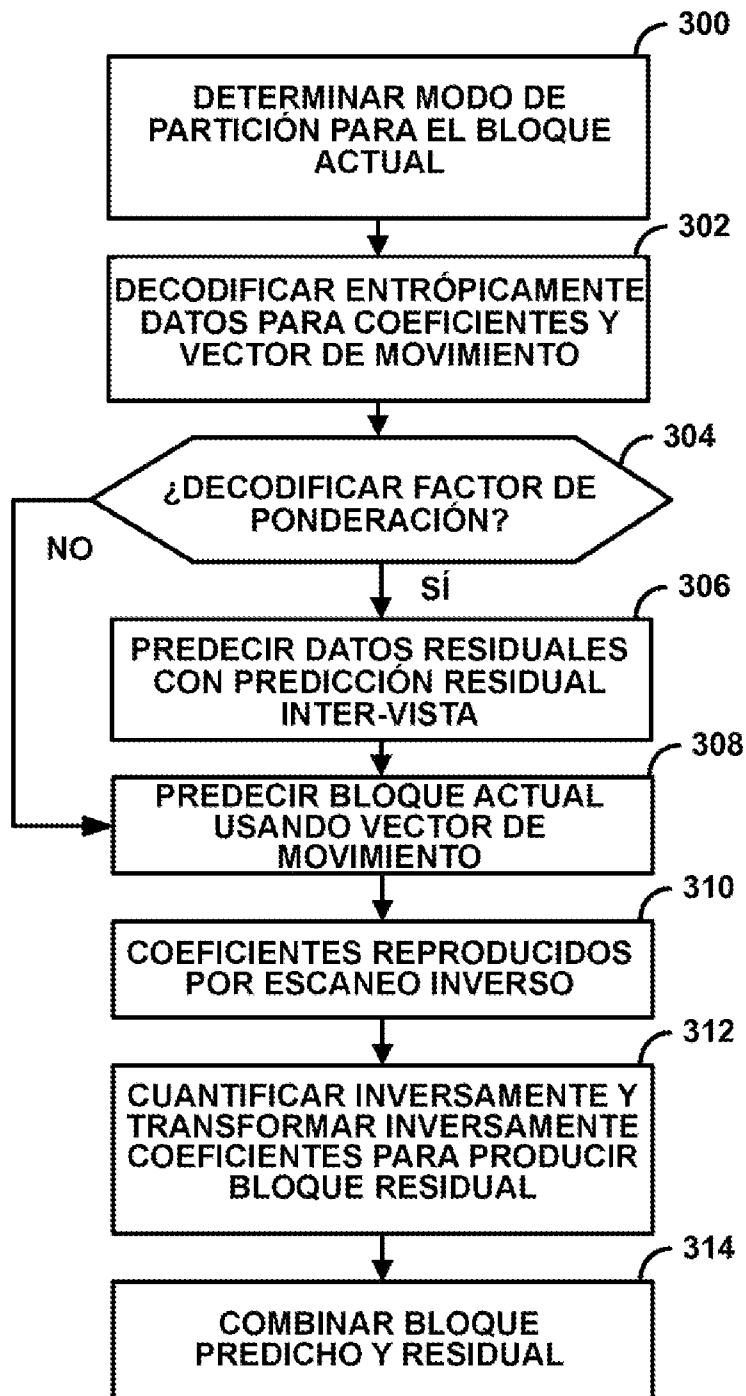


FIG. 18

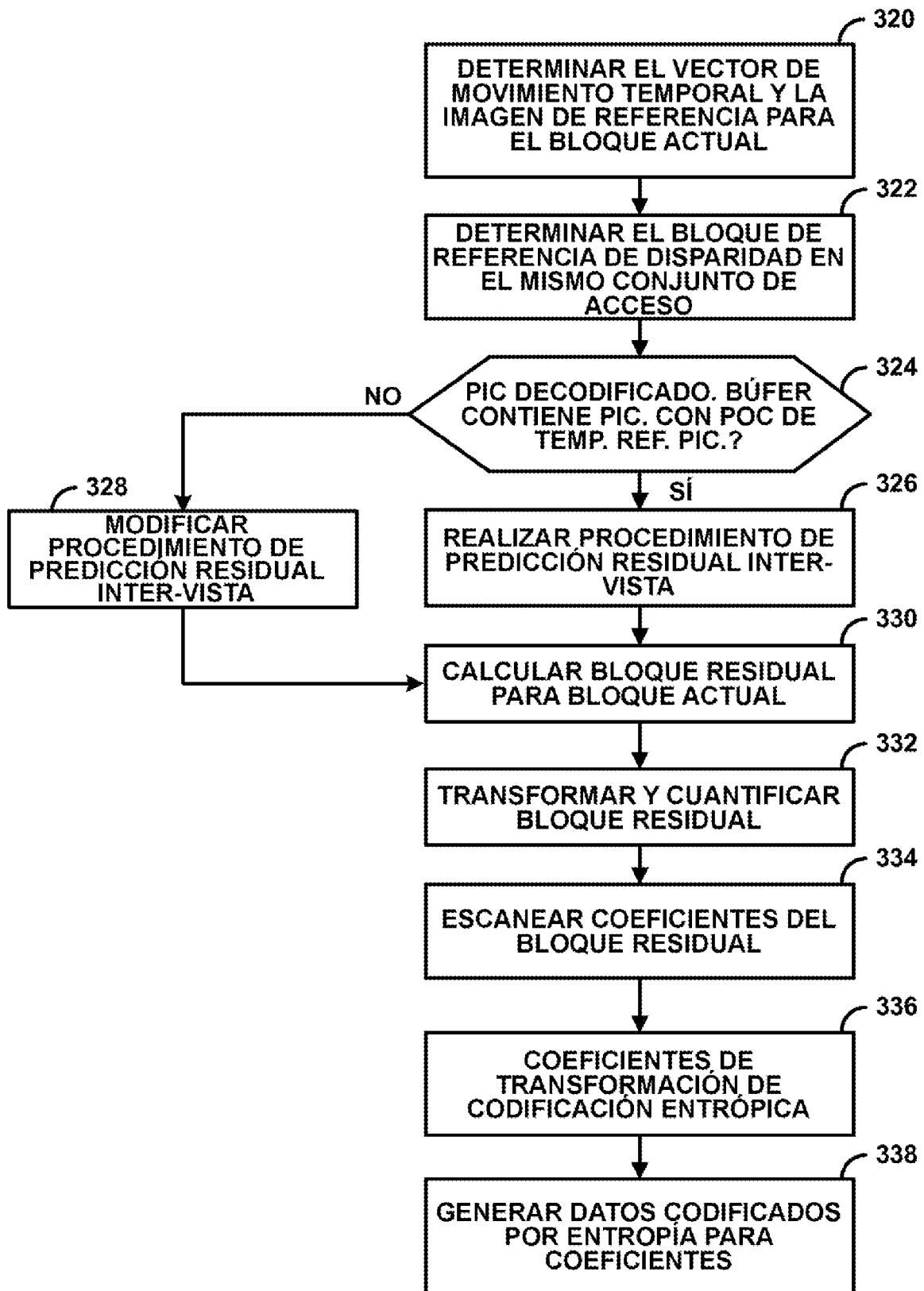


FIG. 19

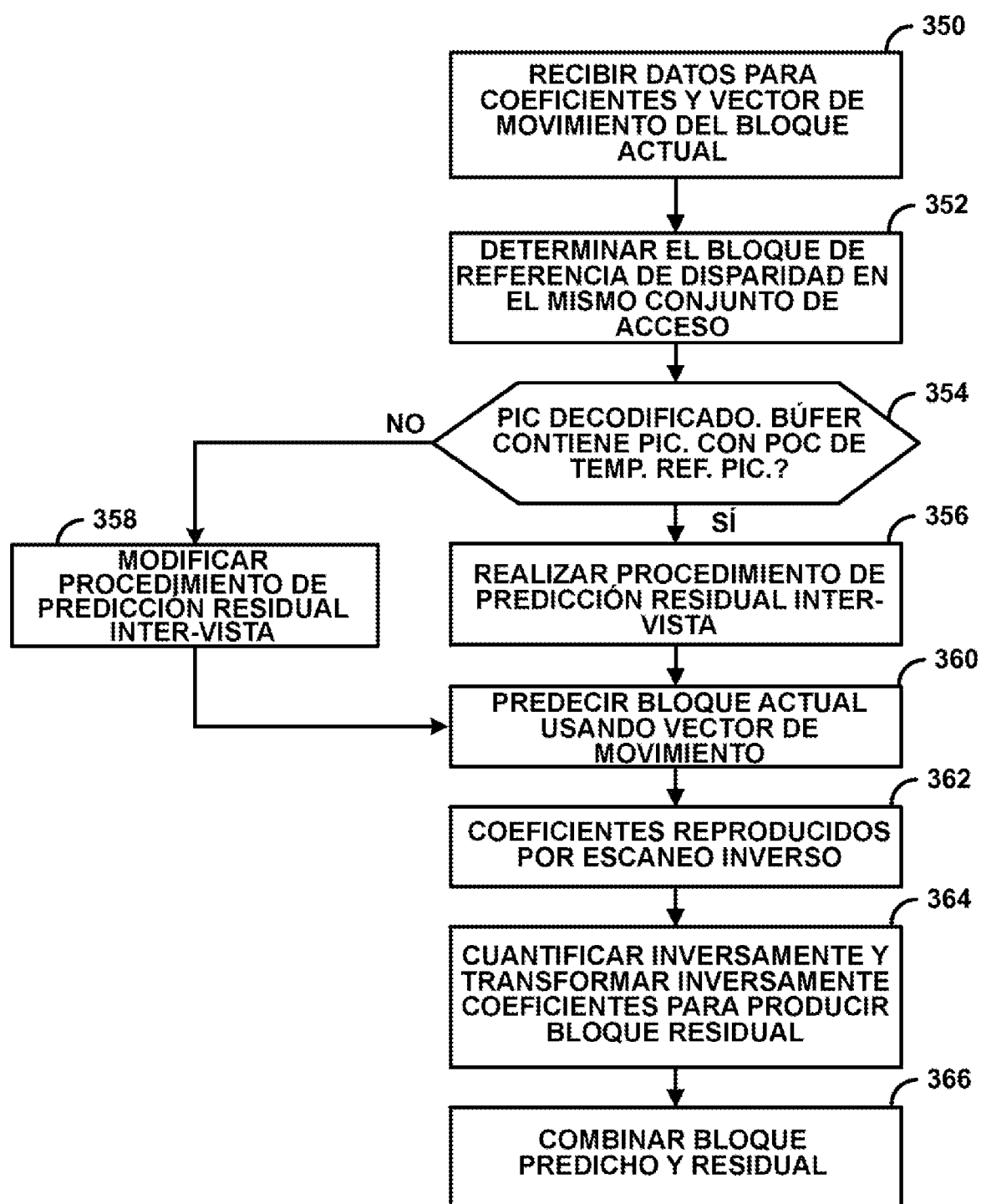


FIG. 20