(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 899 664**

(51) Int. CI.:

H04N 19/597 (2014.01)
H04N 19/70 (2014.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **16.08.2013 PCT/US2013/055403**

(87) Fecha y número de publicación internacional: **20.02.2014 WO14028867**

(96) Fecha de presentación y número de la solicitud europea: **16.08.2013 E 13753248 (7)**

(97) Fecha y número de publicación de la concesión europea: **03.11.2021 EP 2885917**

(54) Título: **Construcción de listas de imágenes de referencia para codificación de vídeo multivistas o 3DV**

(30) Prioridad:

16.08.2012 US 201261684033 P
09.01.2013 US 201361750710 P
17.01.2013 US 201361753822 P
15.08.2013 US 201313968140

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
14.03.2022

(73) Titular/es:

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

(72) Inventor/es:

CHEN, YING;
ZHANG, LI y
RAMASUBRAMONIAN, ADARSH KRISHNAN

(74) Agente/Representante:

ISERN JARA, Jorge

DESCRIPCIÓN

Construcción de listas de imágenes de referencia para codificación de vídeo multivistas o 3DV

Esta solicitud reivindica el beneficio de la Solicitud Provisional de los Estados Unidos Número de serie 61/684,033, presentada el 16 de agosto de 2012, la Solicitud Provisional de los Estados Unidos Número de serie 61/750,710, presentada el 9 de enero de 2013, y la Solicitud Provisional de los Estados Unidos Número de serie 61/753,822, presentada el 17 de enero de 2013.

Campo técnico

Esta divulgación se refiere a la codificación de vídeo.

Antecedentes

Las capacidades de vídeo digital se pueden incorporar en una amplia gama de dispositivos, que incluye televisores digitales, sistemas de difusión digital directa, sistemas de difusión 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, celular o teléfonos por radio satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo en directo, y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de Vídeo Avanzada (AVC), el estándar de Codificación de Vídeo de Alta Eficiencia (HEVC) actualmente en desarrollo, y las extensiones de tales estándares. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficientemente mediante la implementación de tales técnicas de codificación de vídeo.

Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimagen) y/o la predicción temporal (interimagen) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo (por ejemplo, una trama de vídeo o una porción de una trama de vídeo) puede particionarse en bloques de vídeo, que pueden, además, referirse a bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un segmento intracodificado (I) de una imagen se codifican mediante el uso de la predicción espacial con respecto a las muestras de referencia en los bloques vecinos de la misma imagen. Los bloques de vídeo en un segmento 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. Las imágenes se pueden denominar tramas y las imágenes de referencia se pueden denominar tramas de referencia.

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan las 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 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 se pueden transformar del dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, que luego se pueden cuantificar. Los coeficientes de transformación que se cuantifican, dispuestos inicialmente en una matriz bidimensional, pueden escanearse con el fin de producir un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación de entropía para lograr una compresión aún mayor.

La ISO/IEC IS 14496-10 incluye la especificación de la codificación de vídeo avanzada (AVC) y extensiones asociadas, que incluyen la Codificación de Vídeo Escalable (SVC) y la Codificación de Vídeo Multivista (MVC). En la SVC, las capas escalables se identifican mediante un identificador layer_id. En la MVC, las vistas se identifican mediante un identificador view_id. Los elementos de sintaxis asociados con una unidad VCL NAL se codifican de acuerdo con la extensión pertinente.

YING CHEN Y OTROS: "3D-HEVC HLS: reference picture list modification", 100. La REUNIÓN MPEG; 30-4-2012 - 4-5-2012; GINEBRA; (GRUPO DE EXPERTOS EN IMÁGENES EN MOVIMIENTO O ISO/ IEC JTC1/SC29/WG11), número m24944, 25 de abril de 2012 (2012-04-25) divulga los elementos de sintaxis en un encabezado de segmento para insertar imágenes de referencia entre vistas al final de una lista de imágenes de referencia.

Sumario

En general, esta divulgación describe las técnicas para codificar los elementos de sintaxis de encabezados de segmento en base al menos en parte a un paradigma de codificación de vídeo usado para codificar los segmentos, que incluyen los encabezados de segmento. Por ejemplo, los elementos de sintaxis pueden incluir los valores relacionados con la construcción de las listas de imágenes de referencia, por ejemplo, para las imágenes de

referencia entre vistas para la codificación de vídeo tridimensional (3D). Para proporcionar los datos de vídeo usados para los efectos de vídeo 3D, se pueden reproducir dos o más imágenes de una escena de manera simultánea o sustancialmente simultánea, donde las imágenes corresponden a diferentes "vistas" de la escena, es decir, diferentes perspectivas de cámara horizontal. Las técnicas de codificación de vídeo multivista incluyen las técnicas para codificar estas imágenes, por ejemplo, mediante el uso de la predicción entre vistas. Para hacer referencia a una imagen de referencia entre vistas durante la codificación, los codificadores de vídeo pueden usar un índice en una lista de imágenes de referencia. Los codificadores de vídeo y los decodificadores de vídeo se pueden configurar, de acuerdo con las técnicas de esta divulgación, para inicializar y construir las listas de imágenes de referencia, de manera que un valor de índice codificado represente la imagen de referencia correcta en la lista de imágenes de referencia. Es decir, se pueden proporcionar ciertos elementos de sintaxis con valores representativos de cómo construir y/o modificar una lista de imágenes de referencia que son particulares para una extensión de un estándar de codificación de vídeo base cuando un identificador de capa indica que un segmento se incluye en una vista que se codifica de acuerdo con la extensión del estándar de codificación de vídeo base.

En un ejemplo, se proporciona un procedimiento para decodificar los datos de vídeo como se establece en la reivindicación 1.

En otro ejemplo, se proporciona un procedimiento para codificar los datos de vídeo como se establece en la reivindicación 2.

En otro ejemplo, se proporciona un dispositivo para decodificar los datos de vídeo como se establece en la reivindicación 3.

En otro ejemplo, se proporciona un dispositivo para codificar los datos de vídeo como se establece en la reivindicación 4.

En otro ejemplo, un medio de almacenamiento legible por ordenador ha almacenado en el mismo instrucciones que, cuando se ejecutan, hacen que un procesador ejecute el procedimiento de acuerdo con cualquiera de las reivindicaciones de la 1-2.

Los detalles de uno o más ejemplos se establecen en los dibujos acompañantes y en la descripción de más abajo. Otras características, objetivos y ventajas serán evidentes a partir de la descripción y los dibujos, y a partir de las reivindicaciones.

Breve descripción de los dibujos

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 para construir las listas de imágenes de referencia.

La Figura 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar las técnicas para construir las listas de imágenes de referencia.

La Figura 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar las técnicas para construir las listas de imágenes de referencia.

La Figura 4 es un diagrama de flujo que ilustra un procedimiento de ejemplo para construir una lista de imágenes de referencia y usar la lista de imágenes de referencia cuando se codifica una porción de un segmento actual.

La Figura 5 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo multicapa mediante el uso de diferentes estándares de codificación de vídeo, de acuerdo con las técnicas de esta divulgación.

La Figura 6 es un diagrama de flujo que ilustra un procedimiento de ejemplo para decodificar los datos de vídeo de acuerdo con las técnicas de esta divulgación.

Descripción detallada

En general, las técnicas de esta divulgación se dirigen a codificar los elementos de sintaxis en base, al menos en parte, a si un segmento se incluye en una capa que se codifica de acuerdo con un estándar de codificación de vídeo base o en una capa que se codifica de acuerdo con una extensión del estándar de codificación de vídeo base. Por ejemplo, el estándar de codificación de vídeo base puede corresponder a la Codificación de Vídeo de Alta Eficiencia (HEVC), y la extensión puede corresponder a la HEVC multivista (MV-HEVC), a la HEVC tridimensional (3D-HEVC), o a las extensiones escalables HEVC (S-HEVC). Se pueden codificar diferentes capas (por ejemplo, las capas o vistas escalables) de acuerdo con diferentes tipos de paradigmas de codificación. Por ejemplo, una capa base puede codificarse de acuerdo con un estándar de codificación de vídeo base, tal como HEVC, y una capa de mejora (o vista dependiente) puede codificarse de acuerdo con una extensión del estándar de codificación de vídeo base, tal como MV-HEVC, 3D-HEVC, o S-HEVC.

Los elementos de sintaxis para los segmentos en la capa base pueden ajustarse al estándar de codificación de vídeo base (es decir, incluir los elementos de sintaxis que se ajustan al estándar de codificación de vídeo base), mientras que los elementos de sintaxis para los segmentos en la capa de mejora (o vista dependiente) pueden

incluir tanto los elementos de sintaxis que cumplen con el estándar de codificación de vídeo base como los elementos de sintaxis que se ajustan a la extensión del estándar de codificación de vídeo, al asumir que la capa de mejora se codifica mediante el uso de la extensión. Como un ejemplo, los elementos de sintaxis añadidos para una extensión a un estándar de codificación de vídeo base pueden incluir la construcción de la lista de imágenes de referencia y/o los elementos de sintaxis de modificación, como se analiza con mayor detalle más abajo.

La codificación de vídeo (por ejemplo, la codificación o la decodificación) generalmente implica codificar una serie de imágenes al aprovechar las redundancias en las imágenes, ya sean redundancias espaciales o temporales. Las redundancias espaciales pueden aprovecharse mediante el uso de la intrapredicción, de manera que los bloques de una imagen actual pueden predecirse con relación a los píxeles de los bloques vecinos, previamente codificados. Las redundancias temporales pueden aprovecharse mediante el uso de la interpredicción, de manera que los bloques de una imagen actual pueden predecirse con relación a los píxeles de las imágenes codificadas previamente.

Las series de imágenes pueden capturarse o generarse en un cierto orden, que puede ser sustancialmente similar (o idéntico) al orden en el que se visualizarán las imágenes. Sin embargo, el orden en el que se codifican (codifican o decodifican) las series de imágenes no es necesariamente el mismo que el orden en el que se capturan, generan o visualizan las imágenes (generalmente denominado orden de visualización). Esto permite codificar algunas imágenes (por ejemplo, las imágenes B) con referencia tanto a una imagen temporalmente anterior como a una imagen temporalmente posterior, lo que puede producir valores pronosticados que están muy cerca de los valores reales de la imagen actual que se codifica.

Por lo tanto, para la interpredicción de una imagen actual, los codificadores de vídeo (tales como los codificadores de vídeo y los decodificadores de vídeo) pueden generar varias listas de imágenes de referencia. Por ejemplo, la Lista 0 puede incluir las imágenes de referencia que tienen un orden de visualización anterior al de la imagen actual que se codifica, mientras que la Lista 1 puede incluir las imágenes de referencia que tienen un orden de visualización posterior al de la imagen actual que se codifica. De esta manera, las imágenes de referencia que se usarán al codificar la imagen actual (o un segmento de la misma) pueden identificarse mediante el uso de un valor de índice en una de las listas de imágenes de referencia (por ejemplo, la Lista 0 y la Lista 1).

Los estándares de codificación de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU -T H.264 (también conocido como ISO/IEC MPEG-4 AVC), que incluyen sus extensiones de Codificación de Vídeo Escalable (SVC) y Codificación de Vídeo Multivista (MVC). Además, existe un nuevo estándar de codificación de vídeo, específicamente, la Codificación de Vídeo de Alta Eficiencia (HEVC), que se desarrolla por el Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento ISO/IEC (MPEG). El último Borrador de Trabajo (WD) de HEVC, y denominado HEVC WD8 o, simplemente, WD8. El WD8 se describe en Bross y otros, "High Efficiency Vídeo Coding (HEVC) Text Specification Draft 8," documento JCTVC-J1003_d7, Equipo de Colaboración Conjunta sobre Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 10a Reunión: Estocolmo, Suecia, del 11 al 20 de julio de 2012, que, a partir del 14 de octubre de 2013, está disponible desde http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip.

En algunos casos, los datos de vídeo incluyen múltiples capas. Por ejemplo, en la codificación de vídeo escalable para HEVC (S-HEVC), puede haber una capa base y una o más capas de mejora, que pueden incluir datos para mejorar las imágenes de la capa base. Como ejemplo, las imágenes de la capa base pueden tener una resolución espacial relativamente baja, y la(s) capa(s) de mejora pueden incluir los datos para aumentar la resolución espacial de la capa base (y/o las capas de mejora inferiores). Las capas de mejora pueden mejorar la capa base en una o más dimensiones, tales como la resolución espacial, los niveles de calidad de relación señal/ruido (SNR), las profundidades de bits o similares.

Adicional o alternativamente, en la Codificación de Vídeo Multivista para HEVC (por ejemplo, MV-HEVC o 3D-HEVC), puede haber una vista base y una o más vistas dependientes, donde cada vista puede corresponder a una perspectiva de cámara diferente de una escena, por ejemplo, para reproducir los datos de vídeo tridimensionales. Las vistas dependientes pueden codificarse entre vistas con respecto a la vista base (o a una vista no base que es más baja en la jerarquía de vistas). Por lo tanto, cada vista puede considerarse una capa respectiva de los datos de vídeo.

Esta divulgación reconoce que la HEVC puede extenderse aún más en el futuro. S-HEVC, MV-HEVC y 3D-HEVC son las próximas extensiones de HEVC, y pueden agregarse otras extensiones en el futuro. Típicamente, las extensiones de los estándares de vídeo incluyen datos de señalización adicionales. Esta divulgación propone varias técnicas relacionadas con la extensión de HEVC, por ejemplo, para la codificación multicapa (que puede incluir la codificación multivista).

Como ejemplo, esta divulgación propone codificar una capa base de acuerdo con la especificación base de HEVC. De esta manera, los datos señalizados para datos de vídeo en la capa base no necesitan cambiarse con respecto a

la especificación base de HEVC. Una capa base puede definirse como una capa para la cual un identificador de capa (layer_id) tiene un valor de cero. Por lo tanto, se pueden señalar datos adicionales sólo para capas no base, por ejemplo, las capas para las que un identificador de capa tiene un valor distinto de cero.

Como se analizó anteriormente, la codificación de vídeo basada en bloques generalmente incluye la predicción de los bloques de los datos de vídeo, que se pueden realizar mediante el uso de la intrapredicción o interpredicción. Para codificar un bloque mediante el uso de la interpredicción, un codificador de vídeo construye una lista de imágenes de referencia, que incluye las imágenes de referencia potenciales para un segmento que incluye el bloque. Luego, el codificador de vídeo codifica datos para el bloque representativo de un índice en la lista de imágenes de referencia, donde el índice corresponde a la imagen de referencia real que se usa para predecir el bloque. De esta manera, es importante que tanto los codificadores de vídeo como los decodificadores de vídeo construyan la lista de imágenes de referencia de la misma manera.

La construcción de una lista de imágenes de referencia puede incluir la construcción de una lista de imágenes de referencia temporal de acuerdo con las reglas de construcción por defecto a partir de uno o más subconjuntos de un conjunto de imágenes de referencia (donde cada subconjunto puede denominarse como un "subconjunto RPS"). El codificador de vídeo puede entonces reorganizar la lista de imágenes de referencia temporal. Un codificador de vídeo puede determinar una manera apropiada de reorganizar la lista de imágenes de referencia temporal, y un decodificador de vídeo puede usar los datos señalizados para determinar cómo reorganizar la lista de imágenes de referencia temporal. Los codificadores de vídeo también típicamente construyen hasta dos listas de imágenes de referencia por segmento: la lista 0 y la lista 1. Por lo tanto, los codificadores de vídeo construyen hasta dos listas de imágenes de referencia temporales. Una lista de imágenes de referencia puede denominarse RefPicListX, y la lista de imágenes de referencia temporal para RefPicListX puede denominarse RefPicListTempX, donde X es igual a 0 o 1.

De acuerdo con la HEVC WD8, los codificadores de vídeo pueden realizar la inicialización de la lista de imágenes de referencia para crear la Lista 0 y la Lista 1 predeterminadas (si el segmento es un segmento B) en base a tres subconjuntos de conjuntos de imágenes de referencia (RPS): RefPicSetStCurrBefore, RefPicSetStCurrAfter, y RefPicSetLtCurr. Las imágenes de referencia a corto plazo con un orden de salida anterior (posterior) a la imagen actual pueden insertarse primero en la Lista 0 (Lista 1) en orden ascendente de la distancia de recuento de orden de imágenes (POC), luego las imágenes de referencia a corto plazo con el orden de salida posterior (anterior) a la imagen actual pueden insertarse en la Lista 0 (Lista 1) en orden ascendente de la distancia POC, y finalmente las imágenes de referencia a largo plazo pueden insertarse al final.

En términos de RPS, para la Lista 0, las entradas en RefPicSetStCurrBefore pueden insertarse en la lista inicial, seguidas de las entradas en RefPicSetStCurrAfter. Posteriormente, pueden agregarse las entradas en RefPicSetLtCurr, si están disponibles. En la HEVC, el procedimiento anterior se repite (las imágenes de referencia que ya se han agregado a la lista de imágenes de referencia se agregan nuevamente) cuando el número de entradas en una lista es menor que el número objetivo de imágenes de referencia activas (señaladas en un conjunto de parámetros de imagen (PPS) o encabezado de segmento). Cuando el número de entradas es mayor que el número objetivo, la HEVC WD8 especifica que la lista se truncará.

Después de que se haya inicializado una lista de imágenes de referencia, se puede modificar de manera que las imágenes de referencia para la imagen actual se puedan disponer en cualquier orden, que incluye el caso donde una imagen de referencia en particular puede aparecer en más de una posición en la lista, en base a los comandos de modificación de la lista de imágenes de referencia. Cuando un indicador que indica si hay modificaciones presentes o no se establece en uno, se señala un número fijo (igual al número objetivo de entradas en la lista de imágenes de referencia) de comandos, y cada comando inserta una entrada para una lista de imágenes de referencia. Una imagen de referencia puede identificarse en el comando mediante el índice de la lista de imágenes de referencia para la imagen actual derivada de la señalización RPS.

La Tabla 1 más abajo proporciona la sintaxis para una estructura de datos de modificación de la lista de imágenes de referencia:

Tabla 1

ref_pic_list_modification() {	**Descriptor**
ref_pic_list_modification_flag_10	u(1)
if(ref_pic_list_modification_flag_10 && NumPocTotalCurr > 1)	
for(i = 0; i <= num_ref_idx_10_active_minus1;i++)	
list_entry_l0[i]	u(v)
if(slice_type = = B) {	
ref_pic_list_modification_flag_l1	u(1)
if(ref_pic_list_modification_flag_l1 && NumPocTotalCurr > 1)	
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++)	
list_entry_l1[i]	u(v)
}	
}	

La variable NumPocTotalCurr de la Tabla 1 se puede establecer igual a NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr. List_entry_lX [i] (con X igual a 0 o 1) de la Tabla 1 puede especificar el índice de la imagen de referencia en RefPicSetCurrTempListX que se colocará en la posición actual de la lista de imágenes de referencia LX (siendo X 0 o 1). La longitud del elemento de sintaxis list_entry_lX[i] puede definirse como bits Ceil(Log2(NumPocTotalCurr)). El valor de list_entry_lX [i] puede estar en el intervalo de 0 a NumPocTotalCurr - 1, inclusive. Si el elemento de sintaxis list_entry_lX[i] no está presente, puede inferirse que es igual a 0. De esta manera, la Tabla 1 proporciona los elementos de sintaxis que pueden usarse para modificar una lista inicializada.

Los codificadores de vídeo se pueden configurar, de acuerdo con las técnicas de HEVC WD8, para usar los elementos de sintaxis de la Tabla 1 para modificar una lista de imágenes de referencia, de la siguiente manera. El siguiente procedimiento puede invocarse al codificar un segmento P o B. El codificador de vídeo puede establecer la variable NumRpsCurrTempListO igual a Max(num_ref_idx_10_active_minus1+ 1, NumPocTotalCurr) y construir la lista RefPicListTempO de la siguiente manera (donde "#-#" se refiere a una porción correspondiente de HEVC WD8):

```
rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
        for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0;
                rIdx++, i++ )
                RefPicListTemp0[ rIdx ] = RefPicSetStCurrBefore[ i ]
        for( i = 0;  i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0;
                rIdx++, i++ )                                          (8-8)
                RefPicListTemp0[ rIdx ] = RefPicSetStCurrAfter[ i ]
        for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0;
                rIdx++, i++ )
                RefPicListTemp0[ rIdx ] = RefPicSetLtCurr[ i ]
}
```

El codificador de vídeo puede entonces construir la lista RefPicListO de la siguiente manera:

```
for( rIdx = 0; rIdx ≤ num_ref_idx_l0_active_minus1; rIdx++)                (8-9)
        RefPicList0[ rIdx ] = ref_pic_list_modification_flag_l0 ?
                RefPicListTemp0[list_entry_l0[rIdx]] : RefPicListTemp0[rIdx]
```

Donde la estructura de la fórmula "COND ? X : Y" evalúa el valor de COND, y si COND es verdadero, devuelve X, de lo contrario (si COND es falso), devuelve Y.

Cuando el segmento es un segmento B, la variable NumRpsCurrTempList1 puede establecerse igual a Max(num_ref_idx_l1_active_minus1 + 1, NumPocTotalCurr) y el codificador de vídeo puede construir la lista RefPicListTemp1 de la siguiente manera:

```
rIdx = 0
while( rIdx < NumRpsCurrTempList1 ) {
    for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1;
        rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0;  i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList1;
        rIdx++, i++ )                                              (8-10)
        RefPicListTemp1[ rIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList1;
        rIdx++, i++ )
        RefPicListTemp1[ rIdx ] = RefPicSetLtCurr[ i ]
}
```

El codificador de vídeo puede entonces construir RefPicList1, al asumir que el segmento es un segmento B, de la siguiente manera:

```
for( rIdx = 0; rIdx ≤ num_ref_idx_l1_active_minus1; rIdx++)                (8-11)
    RefPicList1[ rIdx ] = ref_pic_list_modification_flag_l1 ?
        RefPicListTemp1[list_entry_l1[rIdx]] : RefPicListTemp1[rIdx]
```

Los estándares de codificación de vídeo pueden ampliarse para soportar otros tipos de datos de vídeo. Por ejemplo, una extensión estéreo/multivista de HEVC se ha estandarizado bajo JCT-3V, sin tener en cuenta la información de profundidad. Las listas de imágenes de referencia de un componente de vista en una vista mejorada pueden señalizarse explícitamente o señalizarse de una manera similar a la extensión MVC de ITU-T H.264/AVC (también denominada H.264/MVC o MVC). Cuando es similar a H.264/MVC, las referencias entre vistas se agregan después de las imágenes de referencia temporal durante la inicialización, de manera que RefPicListTempX (siendo X 0 o 1) contiene las imágenes de referencia entre vistas que siguen a todas las imágenes de referencia temporal. Una vez que se crea RefPicListTempX, la reordenación/modificación de la imagen de referencia se puede hacer únicamente en base a RefPicListTempX, sin la necesidad de saber si una entrada dentro de ella es una referencia entre vistas o no.

Para admitir la codificación de vídeo multicapa (por ejemplo, la predicción entre capas o entre vistas), puede que sea necesario configurar un codificador de vídeo para insertar las imágenes de referencia entre capas (que nuevamente pueden incluir entre vistas) en la lista de imágenes de referencia en una posición particular. En ciertas extensiones, puede ser ventajoso agregar las imágenes de referencia entre capas en una posición, y en otras extensiones, puede ser ventajoso agregar las imágenes de referencia entre capas en una posición diferente. Por lo tanto, un ejemplo de los datos de señalización adicionales son los datos que señalan la posición en la que las imágenes de referencia entre capas deben insertarse en una lista de imágenes de referencia.

Por ejemplo, en un ejemplo, los datos pueden ser señalizados (por ejemplo, en un encabezado de segmento, un conjunto de parámetros de imagen (PPS) o un conjunto de parámetros de secuencia (SPS)) que indica la posición predeterminada en la que se insertan las imágenes de referencia entre capas en una lista de imágenes de referencia inicial. Debe entenderse que estos datos solo se señalarían para una capa no base. En algunos ejemplos, además, la construcción de la lista de imágenes de referencia inicial puede cambiarse en base a los datos señalizados. Alternativamente, pueden cambiarse los datos señalados para reorganizar la lista de imágenes de referencia inicial.

En otro ejemplo, los datos se pueden señalar para una capa no base que indica si las imágenes de referencia entre capas se deben insertar al comienzo de una lista de imágenes de referencia inicial para la lista 0, después de las imágenes de referencia temporal que tienen valores de recuento de orden de imágenes (POC) menores que el valor POC de la imagen actual en la lista 0, después de las imágenes de referencia temporal que tienen valores POC mayores que el valor POC para la imagen actual en la lista 0, o después de las imágenes de referencia a largo plazo (si las hay) en la lista 0. Se pueden señalar datos similares para la lista 1. Las posiciones en la lista 0 pueden ser diferentes a la posición en la lista 1, como lo indican los datos señalados para la lista 0 y la lista 1.

Un posible procedimiento de inicialización de la lista de imágenes de referencia, cuando se codifica un segmento P o B, de la siguiente manera. La variable NumRpsCurrTempListO puede establecerse igual a Max(num_ref_idx_10_active_minus1 + 1, NumPocTotalCurr) y el codificador de vídeo puede construir la lista RefPicListTempO de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
            RefPicListTemp0[ cIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0;  i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )                                              (F-25)
        RefPicListTemp0[ cIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
        RefPicListTemp0[ cIdx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
          RefPicListTemp0[ cIdx ] = RefPicSetIvCurr[ i ]
}
```

El codificador de vídeo puede entonces construir la lista RefPicListO de la siguiente manera:

```
for( cIdx = 0; cIdX ≤ num_ref_idx_l0_active_minus1; cIdx++)          (F-26)
    RefPicList0[ cIdx ] = ref_pic_list_modification_flag_l0 ?
        RefPicListTemp0[list_entry_l0[cIdx]] : RefPicListTemp0[cIdx]
```

Cuando el segmento es un segmento B, el codificador de vídeo puede establecer la variable NumRpsCurrTempList1 igual a Max(num_ref_idx_ll_active_minus1 + 1, NumPocTotalCurr) y construir la lista RefPicListTemp1 de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList1 ) {
    for( i = 0; i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList1;
        cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0;  i < NumPocStCurrBefore && cIdx <
        NumRpsCurrTempList1; cIdx++, i++ )                         (F-27)
        RefPicListTemp1[ cIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList1;
        cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList1;
        cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetIvCurr[ i ]
}
```

Cuando el segmento es un segmento B, el codificador de vídeo puede construir RefPicList1 de la siguiente manera:

```
for( cIdx = 0; cIdX ≤ num_ref_idx_l1_active_minus1; cIdx++)          (F-28)
    RefPicList0[ cIdx ] = ref_pic_list_modification_flag_l1 ?
        RefPicListTemp1[list_entry_l1[cIdx]] : RefPicListTemp1[cIdx]
```

El codificador de vídeo puede establecer la variable NumPocTotalCurr igual a NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr+ NumPocIvCurr. RePlcSetIvCurr se refiere al subconjunto RPS de las imágenes de referencia entre vistas y NumPocIvCurr se refiere al número de imágenes de referencia entre vistas.

El diseño actual de la construcción y modificación de la lista de imágenes de referencia puede sufrir ciertos problemas. Por ejemplo, en muchos escenarios, la imagen de referencia entre vistas no sigue todas las imágenes de referencia temporal, y ni siquiera todas las entradas en RefPicSetStCurrBefore para RefPicListO. Cuando esto sucede, puede ser necesario un ciclo completo de modificación de la lista de imágenes de referencia, lo cual puede costar muchos bits. También puede haber casos en los que una imagen de referencia entre vistas se coloque al final de una lista de imágenes de referencia, lo cual no requiere la modificación de la lista de imágenes de referencia. Esta divulgación describe las técnicas que pueden usarse para superar estos y/u otros problemas asociados con las técnicas actuales de HEVC.

Los codificadores de vídeo se pueden configurar para inicializar y/o modificar las listas de imágenes de referencia de acuerdo con diversas técnicas de esta divulgación. Estas técnicas pueden usarse solas o en cualquier combinación. En un ejemplo, los codificadores de vídeo pueden codificar un valor en un encabezado de segmento representativo de una posición predeterminada de referencias entre vistas en una lista de imágenes de referencia inicial, al estar presentes imágenes de referencia entre vistas consecutivamente en la lista de imágenes de referencia. En otro ejemplo, que puede ser adicional al ejemplo anterior, una lista temporal RefPicListTempX, como una unión de subconjuntos de RPS en un cierto orden, puede cambiarse en base a la posición señalada de las imágenes de referencia entre vistas. Alternativamente, no es necesario cambiar RefPicListTempX, pero en su lugar, se puede cambiar la lista inicial.

En otro ejemplo, los codificadores de vídeo pueden codificar un valor en un encabezado de segmento, conjunto de parámetros de imagen (PPS), o conjunto de parámetros de secuencia (SPS), indicativo de si las imágenes de referencia entre vistas se colocan al comienzo de una lista de imágenes de referencia inicial 0 (RefPicListO) de un segmento, justo después (es decir, inmediatamente después) de las imágenes de referencia temporal con valores de POC menores que el valor de POC actual para RefPicListO, justo después de las imágenes de referencia temporal con valores de POC mayores que el POC actual para RefPicListO, o justo después de las imágenes de referencia a largo plazo, si están presentes. Además, los codificadores de vídeo pueden codificar en el encabezado de segmento, PPS o SPS, ya sea que las referencias entre vistas se coloquen al comienzo de la lista de imágenes de referencia inicial 1 (RefPicList1) de un segmento, justo después de las imágenes de referencia temporal con valores de POC mayores que el POC actual para RefPicList1, justo después de las imágenes de referencia temporal con valores de POC menores que el POC actual para RefPicList1, o justo después de las imágenes de referencia a largo plazo, si están presentes.

Para agregar nuevos elementos de sintaxis para las extensiones a los estándares de codificación de vídeo, tales como las extensiones de HEVC, los codificadores de vídeo pueden codificar dichos elementos de sintaxis como agregados bajo la condición de layer_id (reserved_zero_6bits) no igual a 0. Por lo tanto, estos elementos de sintaxis pueden estar presentes e intercalados con los elementos de sintaxis ya definidos por un estándar de codificación de vídeo base, tal como la especificación base de HEVC.

La Figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo de ejemplo 10 que puede utilizar técnicas para construir las listas de imágenes de referencia. Como se muestra en la Figura 1, el sistema 10 incluye un dispositivo fuente 12 que proporciona datos de vídeo codificados para ser decodificados en un momento posterior mediante un dispositivo de destino 14. En particular, el dispositivo fuente 12 proporciona los datos de vídeo al dispositivo de destino 14 a través de un medio legible por ordenador 16. El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, que incluyen ordenadores de escritorio, ordenadores portátiles (es decir, laptop), tabletas, módulos de conexión, teléfonos móviles tal como los denominados teléfonos "inteligentes", las denominadas smartpads, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión en tiempo real de videos o similares. En algunos casos, el dispositivo fuente 12 y el dispositivo de destino 14 se pueden equipar para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados para decodificarlos 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 fuente 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 fuente 12 transmita datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o alámbrica, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red en base a paquetes, tal como una red de área local, una red de área amplia, o una red global tal como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base, o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo fuente 12 al dispositivo de destino 14.

En algunos ejemplos, los datos codificados pueden salir desde la interfaz de salida 22 hacia un dispositivo de almacenamiento. De manera similar, puede accederse a los datos codificados desde el dispositivo de almacenamiento mediante la 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 que pueda adecuarse para almacenar datos de video 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 video codificado que se genera mediante el dispositivo fuente 12. El dispositivo de destino 14 puede acceder a los datos de vídeo que se almacenan desde el dispositivo de almacenamiento a través de transmisión o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar los datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a la red (NAS), o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, que incluye una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (por ejemplo, DSL, módem por cable, etc.), o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de las mismas.

Las técnicas de esta divulgación no se limitan necesariamente a las aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de video en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como difusión de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de video por Internet, tales como transmisión dinámica adaptativa a través de HTTP (DASH), video digital que se codifica en un medio de almacenamiento de datos, decodificación de video digital que se almacenan en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 se puede configurar para soportar la transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tal como el vídeo en directo, la reproducción de vídeo, la difusión de vídeo, y/o la telefonía de vídeo.

En el ejemplo de la Figura 1, el dispositivo fuente 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 fuente 12 se puede configurar para aplicar las técnicas para construir las listas de imágenes de referencia. En otros ejemplos, un dispositivo fuente y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 12 puede recibir datos de video de una fuente de video externa 18, tal como una cámara externa. Asimismo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema 10 que se ilustra de la Figura 1 es simplemente un ejemplo. Las técnicas para construir listas de imágenes de referencia 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/decodificador de vídeo, típicamente denominado "CODEC". Además, las técnicas de esta divulgación pueden, además, realizarse mediante un preprocesador de vídeo. El dispositivo fuente 12 y el dispositivo de destino 14 son simplemente ejemplos de tales dispositivos de codificación en los que el dispositivo fuente 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 los componentes de codificación y decodificación de vídeo. De ahí que, el sistema 10 puede admitir la transmisión de video unidireccional o bidireccional entre dispositivos de video 12, 14, por ejemplo, para transmisión de video, reproducción de video, difusión de video o videotelefonía.

La fuente de video 18 del dispositivo fuente 12 puede incluir un dispositivo de captura de video, tales como una cámara de video, un archivo de video que contiene video que se captura previamente y/o una interfaz de alimentación de video para recibir video de un proveedor de contenido de video. Como una alternativa adicional, la fuente de video 18 puede generar datos basados en gráficos de ordenador como la fuente de video, o una combinación de video en vivo, video que se archiva y video que se genera por ordenador. En algunos casos, si la fuente de vídeo 18 es una cámara de vídeo, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o videoteléfonos. Sin embargo, como se mencionó anteriormente, las técnicas que se describen en esta divulgación pueden aplicarse a la codificación de vídeo en general, y pueden aplicarse a las aplicaciones inalámbricas y/o por cable. En cada caso, el video que se captura, se precaptura o se genera por ordenador puede codificarse mediante el codificador de video 20. La información de video codificada puede luego salir mediante la interfaz de salida 22 a un medio legible por ordenador 16.

El medio legible por ordenador 16 puede incluir medios transitorios, tal como una transmisión inalámbrica o una transmisión de red por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tal como un disco duro, una unidad flash, un disco compacto, un disco de vídeo digital, disco Blu-ray, u otro medio

legible por ordenador. En algunos ejemplos, un servidor de red (que no se muestra) puede recibir datos de video codificados desde el dispositivo fuente 12 y proporcionar los datos de video codificados al dispositivo de destino 14, por ejemplo, a través de transmisión de red. De manera similar, un dispositivo informático de una instalación de producción media, tal como una instalación de estampado de discos, puede recibir datos de video codificados desde el dispositivo fuente 12 y producir un disco que contenga los datos de video codificados. Por lo tanto, puede entenderse que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de varias formas, en varios ejemplos.

La interfaz de entrada 28 del dispositivo de destino 14 recibe información del medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir la información de sintaxis que se define mediante el codificador de vídeo 20, que se usa además por el decodificador de vídeo 30, que incluye los elementos de sintaxis que describen las características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, los GOP. El dispositivo de visualización 32 muestra los datos de video decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico de emisión de luz (OLED), u otro tipo de dispositivo de visualización.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden operar de acuerdo con un estándar de codificación de vídeo, tal como el estándar de Codificación de Video de Alta Eficiencia (HEVC) actualmente en desarrollo, y pueden ajustarse al Modelo de Prueba HEVC (HM). De manera alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden operar de acuerdo con otros estándares de propiedad o de la industria, tal como el estándar ITU-T H.264, también conocido como MPEG-4, Parte 10, Codificación de Vídeo Avanzada (AVC), o con las extensiones de tales estándares. Las técnicas de esta divulgación, sin embargo, no se limitan a ningún estándar de codificación particular. Otros ejemplos de los estándares de codificación de vídeo incluyen MPEG-2 e ITU-T H.263. Aunque no se muestra en la figura 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden integrar cada uno con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación de audio y de vídeo en un flujo de datos común o en flujos de datos separados. Si corresponde, las unidades MUX-DEMUX se pueden ajustar al protocolo multiplexor ITU H.223, o a otros protocolos tales como el protocolo de datagramas de usuario (UDP).

El estándar ITU-T H.264/MPEG-4 (AVC) se formuló por el Grupo de Expertos en Codificación de Vídeo ITU-T (VCEG) junto con el Grupo de Expertos en Imágenes en Movimiento ISO / IEC (MPEG) como el producto de una asociación colectiva conocida como el Equipo Conjunto de Vídeo (JVT). En algunos aspectos, las técnicas que se describen en esta divulgación pueden aplicarse a los dispositivos que generalmente se ajustan al estándar H.264. El estándar H.264 se describe en la ITU-T Recommendation H.264, Advanced Video Coding for generic audiovisual services, por la Comisión de Estudio del UIT-T, con fecha de marzo de 2005, que puede denominarse en la presente memoria estándar H.264 o especificación H.264, o estándar o especificación H.264/AVC. El Equipo Conjunto de Vídeo (JVT) continúa trabajando en extensiones para H.264/MPEG-4 AVC.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores adecuados, como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados para aplicaciones específicas (ASIC), matriz de puertas lógicas programables en campo (FPGA), lógica discreta, software, hardware, microprograma o cualquiera de sus combinaciones. Cuando las técnicas se implementan de manera parcial en el software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador no transitorio adecuado y ejecutar las instrucciones en el hardware mediante el uso de uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno de los codificadores de vídeo 20 y el descodificador de vídeo 30 puede incluirse en uno o más codificadores o descodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/descodificador combinado (CODEC) en un dispositivo respectivo.

El JCT-VC está trabajando en el desarrollo del estándar HEVC. Los esfuerzos de estandarización de HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba HEVC (HM). El HM presupone diversas capacidades adicionales de los dispositivos de codificación de vídeo con relación a los dispositivos existentes de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación de intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación de intrapredicción.

En general, el modelo de trabajo del HM describe que una trama o imagen de vídeo se puede dividir en una secuencia de bloques de árbol o unidades de codificación más grandes (LCU) que incluyen tanto las muestras de luma como de croma. Los datos de sintaxis dentro de un flujo de bits pueden definir un tamaño para los LCU, que es una unidad de codificación más grande en términos de número de píxeles. Un segmento incluye una serie de bloques de árbol consecutivos en el orden de codificación. Una trama o una imagen de vídeo se pueden dividir en una o más secciones. Cada bloque de árbol se puede dividir en unidades de codificación (CU) de acuerdo con un árbol cuádruple. En general, una estructura de datos de árbol cuaternario incluye un nodo por CU, con un nodo raíz correspondiente al bloque de árbol. Si una CU se divide en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU.

Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar los datos de sintaxis para la correspondiente CU. Por ejemplo, un nodo en el árbol cuaternario puede incluir una bandera de división, que indica si la CU correspondiente al nodo se divide en sub-CU. Los elementos de sintaxis para una CU pueden definirse recursivamente y pueden depender de si la CU se divide en sub-CU. Si una CU no se divide más, se refiere a ella como una CU hoja. En esta divulgación, cuatro sub-CU de una CU hoja también se denominarán CU hojas incluso si no hay una división explícita de la CU hoja original. Por ejemplo, si una CU con un tamaño de 16x16 no se divide más, las cuatro sub-CU de 8x8 también puede referirse como CU hoja, aunque la CU de 16x16 nunca se dividió.

Una CU tiene un propósito similar a un macrobloque del estándar H.264, excepto que una CU no tiene una distinción de tamaño. Por ejemplo, un bloque de árbol puede dividirse en cuatro nodos hijo (también denominados sub-CU), y cada nodo hijo puede ser a su vez un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final no dividido, denominado nodo hoja del árbol cuaternario, comprende un nodo de codificación, también denominado CU hoja. Los datos de sintaxis asociados con un flujo de bits codificado pueden definir un número máximo de veces que un bloque de árbol se puede dividir, denominado profundidad máxima de CU, y también pueden definir un tamaño mínimo de los nodos de codificación. En consecuencia, un flujo de bits también puede definir una unidad de codificación más pequeña (SCU). Esta divulgación usa el término "bloque" para referirse a cualquier CU, PU o TU, en el contexto de HEVC, o a estructuras de datos similares en el contexto de otros estándares (por ejemplo, macrobloques y subbloques de los mismos en H.264/AVC).

Una CU incluye un nodo de codificación y las unidades de predicción (PU) y las unidades de transformación (TU) asociadas con el nodo de codificación. Un tamaño de la CU corresponde al tamaño del nodo de codificación y debe ser de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol con un máximo de 64x64 píxeles o superior. Cada CU puede contener una o más PU y una o más TU. Los datos de sintaxis asociados con una CU pueden describir, por ejemplo, la partición de la CU en una o más PU. Los modos de particionamiento pueden diferir entre si la CU es codificada en modo directo o en modo de salto, codificada en modo de intrapredicción, o codificada en modo de interpredicción. Las PU se pueden dividir para que sean de forma no cuadrada. Los datos de sintaxis asociados con una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuádruple. Una TU puede ser de forma cuadrada o no cuadrada (por ejemplo, rectangular).

El estándar HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU se basa usualmente en el tamaño de las PU dentro de una CU determinada definida para una LCU particionada, aunque esto puede no ser siempre así. Las TU usualmente son del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU se pueden subdividir en unidades más pequeñas utilizando una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos hoja del RQT pueden referirse como unidades de transformación (TU). Los valores de diferencia de píxeles asociados con las TU se pueden transformar para producir coeficientes de transformación, que se pueden cuantificar.

Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa un área espacial correspondiente a toda o una porción de la CU y puede incluir los datos para recuperar una muestra de referencia para la PU. Además, una PU incluye los datos relacionados con la predicción. Por ejemplo, cuando la PU se codifica intramodo, los datos para la PU pueden incluirse en un árbol cuaternario residual (RQT), que puede incluir los datos que describen un modo de intrapredicción para una TU correspondiente a la PU. Como otro ejemplo, cuando la PU se codifica intramodo, la PU puede incluir los datos que definen uno o más vectores de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento, y/o una lista de imágenes de referencia (por ejemplo, Lista 0, Lista 1, o Lista C) para el vector de movimiento.

Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Las unidades de transformación se pueden especificar mediante el uso de un RQT (también denominado estructura de árbol cuaternario TU), como se analizó anteriormente. Por ejemplo, un indicador de división puede indicar si una CU hoja se divide en cuatro unidades de transformación. Entonces, cada unidad de transformación puede dividirse en más sub-TU. Cuando una TU no se divide más, puede denominarse TU hoja. Generalmente, para la intracodificación, todas las TU hoja que pertenecen a una CU hoja comparten el mismo modo de intrapredicción. Es decir, el mismo modo de intrapredicción se aplica generalmente para calcular los valores pronosticados para todas las TU de una CU hoja. Para la intracodificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja mediante el uso del modo de intrapredicción, como una diferencia entre la porción de la CU correspondiente a la TU y al bloque original. Una TU no se limita necesariamente al tamaño de una PU. Por lo tanto, las TU pueden ser más grandes o más pequeñas que una PU. Para la intracodificación, una PU puede colocarse con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder al tamaño de la CU hoja correspondiente.

Además, las TU de las CU hoja también se pueden asociar con las respectivas estructuras de datos de árbol cuaternario, denominadas árboles cuaternarios residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuaternario que indica cómo se particiona la CU hoja en las TU. El nodo raíz de un árbol cuaternario TU corresponde generalmente a una CU hoja, mientras que el nodo raíz de un árbol cuaternario CU corresponde generalmente a un bloque de árbol (o LCU). Las TU del RQT que no están divididas se denominan TU hoja. En general, esta divulgación usa los términos CU y TU para referirse a CU hoja y TU hoja, respectivamente, a menos que se indique lo contrario.

Una secuencia de vídeo usualmente incluye una serie de cuadros o imágenes de vídeo. Un grupo de imágenes (GOP) de manera general comprende una serie de una o más de las imágenes de vídeo. Un GOP puede incluir los datos de sintaxis en un encabezado del GOP, un encabezado de una o más de las imágenes, o en cualquier otro lugar, que describa una serie de imágenes incluidas en el GOP. Cada segmento de una imagen puede incluir los datos de sintaxis del segmento que describen un modo de codificación para el segmento respectivo. El codificador de vídeo 20 usualmente opera en los bloques de vídeo dentro de los segmentos de vídeo individuales para codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables, y pueden diferir en tamaño de acuerdo con un estándar de codificación específico.

Como un ejemplo, el HM soporta la predicción en varios tamaños de PU. Asumiendo que el tamaño de una CU particular es 2Nx2N, el HM soporta la intrapredicción en los tamaños de PU de 2Nx2N o NxN, y la interpredicción en los tamaños de PU simétricos de 2Nx2N, 2NxN, Nx2N o NxN. El HM también soporta particiones asimétricas para la interpredicción en los tamaños de PU de 2NxnU, 2NxnD, nLx2N, y nRx2N. En la partición asimétrica, una dirección de una CU no se particiona, mientras que la otra dirección se particiona en 25 % y 75 %. La parte de la CU correspondiente a la partición del 25 % se indica mediante una "n" seguida de una indicación de "Arriba", "Abajo", "Izquierda", o "Derecha". Por tanto, por ejemplo, "2NxnU" se refiere a una CU de 2Nx2N que se particiona de manera horizontal con una PU de 2Nx0.5N en la parte superior y una PU de 2Nx1.5N en la parte inferior.

En esta divulgación, "NxN" y "N por N" se pueden utilizar indistintamente para referirse a las dimensiones de los píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16x16 píxeles o 16 por 16 píxeles. En general, un bloque de 16x16 tendrá 16 píxeles en una dirección vertical (y = 16) y 16 píxeles en una dirección horizontal (x = 16). Del mismo modo, un bloque NxN de manera general tiene N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden organizar en filas y columnas. Además, los bloques no necesitan tener necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender NxM píxeles, donde M no es necesariamente igual a N.

Después de la codificación intrapredictiva o interpredictiva utilizando las PU de una CU, el codificador de vídeo 20 puede calcular los datos residuales para las TU de la CU. Las PU pueden comprender los datos de sintaxis que describen un procedimiento o modo de generar los datos de píxeles predictivos en el dominio espacial (también denominado dominio de píxeles) y las TU pueden comprender los coeficientes en el dominio de transformación después la aplicación de una transformada, por ejemplo, una transformada de coseno discreta (DCT), una transformada de entero, una transformada de ondícula, o una transformada conceptualmente similar a los datos de vídeo residuales. Los datos residuales pueden corresponder a las diferencias de píxeles entre los píxeles de la imagen no codificada y los valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU que incluyen los datos residuales para la CU, y luego transformar las TU para producir los coeficientes de transformación para la CU.

Después de cualquier transformación para producir los coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación de manera general se refiere a un procedimiento en el que los coeficientes de transformación se cuantifican para de manera posible reducir la cantidad de datos utilizados para representar los coeficientes, proporcionando compresión adicional. El procedimiento de cuantificación puede reducir la profundidad de bits que se asocia con algunos o todos los coeficientes. Por ejemplo, un valor de bits *n* se puede redondear a un valor de bits m durante la cuantificación, donde *n* es mayor que *m.*

Después de la cuantificación, el codificador de vídeo puede escanear los coeficientes de transformación, al producir un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantificados. El escaneo puede diseñarse para colocar los coeficientes de energía más alta (y por lo tanto de frecuencia más baja) en la parte delantera de la matriz y para colocar los coeficientes de energía más baja (y por lo tanto de frecuencia más alta) en la parte posterior de la matriz. En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformación cuantificados para producir un vector serializado que puede codificarse por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneo adaptativo. Después de escanear los coeficientes de transformación cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en

la sintaxis (SBAC), la codificación por Entropía de Particiones de Intervalos de Probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede codificar por entropía los elementos de sintaxis asociados con los datos de vídeo codificados para su uso por el decodificador de vídeo 30 al decodificar los datos de vídeo.

Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto se puede relacionar a, por ejemplo, si los valores vecinos del símbolo son o no de valor cero. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para transmita un símbolo a transmitir. Las palabras de código en VLC se pueden construir de manera que los códigos relativamente más cortos correspondan a los símbolos más probables, mientras que los códigos más largos correspondan a los símbolos menos probables. De esta manera, el uso de VLC puede lograr un ahorro de bits con respecto a, por ejemplo, mediante el uso de palabras de código de igual longitud para cada símbolo a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.

De acuerdo con las técnicas de esta divulgación, se puede configurar un codificador de vídeo (por ejemplo, un codificador de vídeo 20 o un decodificador de vídeo 30) para utilizar diferentes conjuntos de elementos de sintaxis para un encabezado de segmento de un segmento en base a si el segmento está en una capa base o en una capa de mejora. Es decir, al asumir que los datos de vídeo de la capa base se codifican de acuerdo con un estándar de codificación de vídeo base (por ejemplo, HEVC) y que los datos de vídeo de la capa de mejora se codifican de acuerdo con una extensión del estándar de codificación de vídeo base (por ejemplo, MV- HEVC, 3D-HEVC, S-HEVC, o similares), se pueden proporcionar elementos de sintaxis adicionales en los encabezados de segmento de los segmentos en la capa de mejora, con relación a los elementos de sintaxis en los encabezados de segmento de los segmentos en la capa base. Por lo tanto, el codificador de vídeo se puede configurar para utilizar un primer conjunto de elementos de sintaxis para los encabezados de segmento de los segmentos en base a si los segmentos están en una capa base o en una capa de mejora.

El codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para codificar un identificador de capa en un encabezado de segmento para un segmento, donde el identificador de capa (por ejemplo, layer_id) tiene un valor representativo de la capa en la que está presente el segmento. En particular, el codificador de vídeo 20 puede determinar si los datos de vídeo se están codificando para una capa base o una capa de mejora, y codifican un valor para el identificador de capa en consecuencia. El decodificador de vídeo 30, por otro lado, puede configurarse para determinar si un segmento es parte de una capa base o una capa de mejora en base a un valor decodificado para el identificador de capa. Un valor de cero para el identificador de capa generalmente indica que la capa correspondiente es la capa base, mientras que un valor distinto de cero para el identificador de capa generalmente indica que la capa correspondiente es una capa no base, por ejemplo, una capa de mejora (o una vista dependiente, para la codificación de vídeo multivista).

Por lo tanto, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para codificar un primer conjunto de elementos de sintaxis de un encabezado de segmento para un segmento de acuerdo con un estándar de codificación de vídeo base (por ejemplo, HEVC) cuando el identificador de capa tiene un valor de cero (u otro valor que indique que la capa correspondiente es la capa base). Igualmente, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para codificar el primer conjunto de elementos de sintaxis de acuerdo con el estándar de codificación de vídeo base, así como también un segundo conjunto de elementos de sintaxis de acuerdo con una extensión del estándar de codificación de vídeo base (por ejemplo, MV-HEVC, 3D-HEVC, S-HEVC o similares) cuando el identificador de capa tiene un valor distinto de cero (u otro valor que indique que la capa correspondiente no es la capa base). Cuando se combinan, el primer y segundo conjunto de elementos de sintaxis pueden representar el conjunto completo de elementos de sintaxis para un encabezado de segmento de un segmento en la capa de mejora correspondiente al identificador de capa.

Como se analizó anteriormente, un conjunto de elementos de sintaxis que pueden codificarse para una capa de mejora (o vista dependiente) es un conjunto de uno o más elementos de sintaxis para la inicialización y/o modificación de las listas de imágenes de referencia. El codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para inicializar las listas de imágenes de referencia de acuerdo con cualquiera o todas las técnicas de esta divulgación, solas o en cualquier combinación. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para codificar un encabezado de segmento de acuerdo con la Tabla 2 más abajo:

Tabla 2

slice_header() {	**Descriptor**
first_slice_in_pic_flag	u(1)
...	
if(slice_header_extension_present_flag) { // siempre debe ser verdadero en MV-HEVC	
slice_header_extension_length	ue(v)
if(slice_type!= I_SLICE)	
inter view ref start position	ue(v)
...	
}	
byte_alignment( )	
}	

El encabezado de segmento en el ejemplo de la Tabla 2 incluye un elemento de sintaxis adicional, inter_view_ref_start_position, con relación al encabezado de segmento convencional de, por ejemplo, HEVC WD8. Aunque no se muestra en la Tabla 2, en algunos ejemplos, este elemento de sintaxis adicional solo estaría presente en los encabezados de segmento que tengan un identificador de capa que indique que el segmento ocurre en una capa no base (o vista no base), por ejemplo, un layer_id no igual a cero. En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar los valores para inter_view_ref start_position en otras posiciones del encabezado de segmento, en lugar de seguir segmento_header_extension_length. En el ejemplo de la Tabla 2, inter_view_ref_start_position puede especificar la posición inicial de las imágenes de referencia entre vistas en la lista de imágenes de referencia 0 después de la inicialización de la lista de imágenes de referencia. Inter_view_ref_start_position puede tener un valor en el intervalo de 0 a min(num_ref_idx_10_active_minus1, NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr), inclusive. Alternativamente, inter_view_ref_start_position puede tener un valor en el intervalo de 0 a num_ref_idx_10_active_minus1, inclusive.

En el ejemplo de la Tabla 2, inter_view_ref_start_position se codifica como ue(v), es decir, como un entero sin signo mediante el uso de la codificación Exp-Golomb. En otros ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar en su lugar inter_view_ref_start_position como u(v), es decir, como un entero sin signo mediante el uso de un número variable de bits. El número de bits usado puede depender del número de bits usado para codificar otros elementos de sintaxis en el encabezado de segmento. En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar además un valor para inter_view_ref_start_position_11, para especificar la posición de inicio de las imágenes de referencia entre vistas en la lista de imágenes de referencia 1 después de la inicialización de la lista de imágenes de referencia.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden construir RefPicListTempO de una manera diferente a HEVC WD8 y H.264/MVC. En particular, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden colocar las imágenes de referencia entre vistas en las posiciones relacionadas con inter_view_ref_start_position. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para realizar el siguiente procedimiento de inicialización para las listas de imágenes de referencia, por ejemplo, al codificar un encabezado de segmento P o B. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden establecer la variable NumRpsCurrTempListO igual a Max(num_ref_idx_10_active_minus1 + 1, NumPocTotalCurr) y construir RefPicListTempO de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList0 - NumPocIvCurr ) {
        for( i = 0; i < NumPocStCurrBefore && cIdx < NumRpsCurrTempList0;
                cIdx++, i++ )
                RefPicListTemp0[ cIdx ] = RefPicSetStCurrBefore[ i ]
        for( i = 0;  i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList0;
                cIdx++, i++ )                                          (F-25)
                RefPicListTemp0[ cIdx ] = RefPicSetStCurrAfter[ i ]
        for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList0;
                cIdx++, i++ )
                RefPicListTemp0[ cIdx ] = RefPicSetLtCurr[ i ]
}
for(cIdx = NumRpsCurrTempList0-1; cIdx >=inter_view_ref_start_position+
        NumPocIvCurr; cdx--)
        RefPicListTemp0[ cIdx ] = RefPicSetIvCurr[cIdx - NumPocIvCurr]
for (i=0; i< NumPocIvCurr; i++)
        RefPicListTemp0[inter_view_ref_start_position+i] = RefPicSetIvCurr[i]
```

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden entonces construir la lista RefPicListO de la siguiente manera:

```
for( cIdx = 0; cIdX ≤ num_ref_idx_l0_active_minus1; cIdx++)                (F-26)
    RefPicList0[ cIdx ] = ref_pic_list_modification_flag_l0 ?
        RefPicListTemp0[list_entry_l0[cIdx]] : RefPicListTemp0[cIdx]
```

Cuando el segmento es un segmento B, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden establecer la variable NumRpsCurrTempList1 igual a Max(num_ref_idx_11_active_minusl + 1, NumPocTotalCurr) y construir la lista RefPicListTemp1 de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList1 ) {
        for( i = 0; i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList1;
                cIdx++, i++ )
                        RefPicListTemp1[ cIdx ] = RefPicSetStCurrAfter[ i ]
        for( i = 0;  i < NumPocStCurrBefore && cIdx <
                NumRpsCurrTempList1; cIdx++, i++ )                     (F-27)
                RefPicListTemp1[ cIdx ] = RefPicSetStCurrBefore[ i ]
        for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList1;
                cIdx++, i++ )
                RefPicListTemp1[ cIdx ] = RefPicSetLtCurr[ i ]
        for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList1;
                cIdx++, i++ )
                RefPicListTemp1[ cIdx ] = RefPicSetIvCurr[ i ]
}
```

Cuando el segmento es un segmento B, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden construir RefPicList1 de la siguiente manera:

```
for( cIdx = 0; cIdX ≤ num_ref_idx_l1_active_minus1; cIdx++)                (F-28)
    RefPicList0[ cIdx ] = ref_pic_list_modification_flag_l1 ?
        RefPicListTemp1[list_entry_l1[cIdx]] : RefPicListTemp1[cIdx]
```

Alternativamente, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para realizar el siguiente procedimiento de codificación. En este ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden construir RefPicLIstTemp0 de una manera que las referencias entre vistas sigan todavía a las referencias temporales. Sin embargo, cuando se construye la lista de imágenes de referencia inicial, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden colocar las referencias entre vistas en posiciones anteriores en base a inter_view_ref_start_position. Inter_view_ref_start_position puede tener un valor en el intervalo de 0 a M = min (num_ref_idx_10_active_minus1, NumPocStCurrBefore-1), inclusive. Alternativamente, M puede establecerse igual a min (num_ref_idx_10_active_minus1, NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr+ NumPocIvCurr-1). Como otra alternativa, M puede establecerse igual a min (num_ref_idx_10_active_minus1, NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr -2). En otra alternativa más, M puede establecerse igual a min (num_ref_idx_10_active_minus1, NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr -1). Alternativamente, inter_view_ref_start_position puede codificarse como u(v), con bits Ceil(Log2(M+1)).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden inicializar la lista de imágenes de referencia, cuando se codifica un encabezado de segmento P o B, como se describe más abajo. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden establecer la variable NumRpsCurrTempListO igual a Max(num_ref_idx_10_active_minus1 + 1, NumPocTotalCurr) y construir la lista RefPicListTempO de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
        RefPicListTemp0[ cIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0;  i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )                                                  (F-25)
        RefPicListTemp0[ cIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
        RefPicListTemp0[ cIdx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
        RefPicListTemp0[ cIdx ] = RefPicSetIvCurr[ i ]
}
NumTemporalRef = NumPocStCurrBefore + NumPocStCurrAfter +
    NumPocLtCurr;
for ( i=0; i < inter_view_ref_start_position; i++ )
    list_default_entry_l0 [i] = i;
for ( j=0; j< NumPocIvCurr; j++,i++)
    list_default_entry_l0 [i] = NumTemporalRef +j;
for ( ; i< NumRpsCurrTempList0; i++)
    list_default_entry_l0 [i]=i - NumPocIvCurr;
```

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden entonces construir RefPicListO de la siguiente manera:

```
for( cIdx = 0; cIdX ≤ num_ref_idx_l0_active_minus1; cIdx++)                (F-26)
        RefPicList0[ cIdx ] = ref_pic_list_modification_flag_l0 ?
                RefPicListTemp0[list_entry_l0[cIdx]] :
                RefPicListTemp0[list_default_entry_l0[cIdx]]
```

Cuando el segmento es un segmento B, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden establecer la variable NumRpsCurrTempList1 igual a Max(num_ref_idx_11_active_minusl + 1, NumPocTotalCurr) y construir la lista RefPicListTemp1 de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList1 ) {
        for( i = 0; i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList1;
                cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetStCurrAfter[ i ]
        for( i = 0;  i < NumPocStCurrBefore && cIdx <
                NumRpsCurrTempList1; cIdx++, i++ )                          (F-27)
        RefPicListTemp1[ cIdx ] − RefPicSetStCurrBefore[ i ]
        for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList1;
                cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetLtCurr[ i ]
        for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList1;
                cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetIvCurr[ i ]
}
```

Cuando el segmento es un segmento B, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden construir la lista RefPicList1 de la siguiente manera:

```
for( cIdx = 0; cIdX ≤ num_ref_idx_l1_active_minus1; cIdx++)                (F-28)
        RefPicList0[ cIdx ] = ref_pic_list_modification_flag_l1 ?
                RefPicListTemp1[ list_entry_l1[cIdx]] : RefPicListTemp1[cIdx]
```

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden establecer la variable NumPocTotalCurr igual a NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr+ NumPocIvCurr. En este ejemplo, RePIcSetIvCurr es el subconjunto RPS de las imágenes de referencia entre vistas.

Alternativamente, en los diversos ejemplos anteriores, se pueden aplicar procesos de decodificación similares a la lista de imágenes de referencia 1, con la posición de inicio de las imágenes de referencia entre vistas en el encabezado de segmento.

En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar un indicador de 2 bits inter_view_pos_idc que indica dónde añadir las referencias entre vistas en RefPicListTempO. La Tabla 3 más abajo resume un ejemplo de los valores de inter_view_pos_idc y la semántica para cada posible valor binario de este indicador de dos bits:

Tabla 3

inter_view_pos_idc	0	1	2	3
Posiciones RPS entre vistas	Primer subconjunto RPS en RefPicListTempO	Subconjunto RPS entre vistas que se agregará en RefPicListTemp0 después de RefPicSetStCurrBefore	Subconjunto RPS entre vistas que se agregará en RefPicListTe mp0 después de RefPicSetStC urrAfter	Subconjunto RPS entre vistas que se agregará al final

El codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para realizar el siguiente procedimiento para inicializar RefPicListTempO.

```
cIdx = 0
while( cIdx < NumRpsCurrTempList0 ) {
    if (inter_view_pos_idc ==0)
        for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList0;
            cIdx++, i++ )
                RefPicListTemp0[ cIdx ] = RefPicSetIvCurr[ i ]
    for( i = 0; i < NumPocStCurrBefore && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
        RefPicListTemp0[ cIdx ] = RefPicSetStCurrBefore[ i ]
    if (inter_view_pos_idc ==1)
        for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList0;
            cIdx++, i++ )
            RefPicListTemp0[ cIdx ] = RefPicSetIvCurr[ i ]
    for( i = 0;  i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )                                                    (F-25)
        RefPicListTemp0[ cIdx ] = RefPicSetStCurrAfter[ i ]
    if (inter_view_pos_idc ==2)
      for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList0;
          cIdx++, i++ )
          RefPicListTemp0[ cIdx ] = RefPicSetIvCurr[ i ]
   for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList0; cIdx++,
      i++ )
      RefPicListTemp0[ cIdx ] = RefPicSetLtCurr[ i ]
   if (inter_view_pos_idc ==3)
      for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList0;
          cIdx++, i++ )
          RefPicListTemp0[ cIdx ] = RefPicSetIvCurr[ i ]
}
```

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar inter_view_pos_idc en el encabezado de segmento, conjunto de parámetros de imagen (PPS), conjunto de parámetros de secuencia (SPS), conjunto de parámetros de adaptación (APS), o conjunto de parámetros de vídeo (VPS).

En algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden colocar originalmente las imágenes de referencia entre vistas después de RefPicSetStCurrBefore. Por ejemplo, al codificar un encabezado de segmento P o B, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden establecer la variable

NumRpsCurrTempListO igual a Max(num_ref_idx_10_active_minus1 + 1, NumPocTotalCurr) y construir la lista RefPicListTempO de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
        RefPicListTemp0[ cIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
        RefPicListTemp0[ cIdx ] = RefPicSetIvCurr[ i ]
    for( i = 0;  i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )                                              (F-25)
        RefPicListTemp0[ cIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
        RefPicListTemp0[ cIdx ] = RefPicSetLtCurr[ i ]
}
    for ( i=0; i < inter_view_ref_start_position; i++ )
        list_default_entry_l0 [i] = i;
    for ( j=0; j< NumPocIvCurr; j++,i++)
        list_default_entry_l0 [i] = NumPocStCurrBefore +j;
    for ( ; i< NumRpsCurrTempList0; i++)
        list_default_entry_l0 [i]=i - NumPocIvCurr;
```

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden entonces construir la lista RefPicListO de la siguiente manera:

```
for( cIdx = 0; cIdX ≤ num_ref_idx_l0_active_minus1; cIdx++)          (F-26)
    RefPicList0[ cIdx ] = ref_pic_list_modification_flag_l0 ?
        RefPicListTemp0[ list_entry_l0[ cIdx ] ] :
        RefPicListTemp0[list_default_entry_l0[cIdx] ]
```

Cuando el segmento es un segmento B, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden establecer la variable NumRpsCurrTempList1 igual a Max(num_ref_idx_11_active_minusl + 1, NumPocTotalCurr) y construir la lista RefPicListTemp1 de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList1 ) {
    for( i = 0; i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList1;
        cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0;  i < NumPocStCurrBefore && cIdx <
        NumRpsCurrTempList1; cIdx++, i++ )                    (F-27)
        RefPicListTemp1[ cIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList1;
        cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList1;
        cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetIvCurr[ i ]
}
```

Alternativamente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden construir la lista RefPicListTemp1 de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList1 ) {
    for( i = 0; i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList1;
        cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0;  i < NumPocStCurrBefore && cIdx <
        NumRpsCurrTempList1; cIdx++, i++ )                    (F-27)
        RefPicListTemp1[ cIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocIvCurr && cIdx < NumRpsCurrTempList1;
        cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetIvCurr[ i ]
    for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList1;
        cIdx++, i++ )
        RefPicListTemp1[ cIdx ] = RefPicSetLtCurr[ i ]
}
```

Cuando el segmento es un segmento B, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden construir la lista RefPicList1 de la siguiente manera:

```
for( cIdx = 0; cIdX ≤ num_ref_idx_l1_active_minus1; cIdx++)          (F-28)
    RefPicList0[ cIdx ] = ref_pic_list_modification_flag_l1 ?
        RefPicListTemp1[list_entry_l1[cIdx]] : RefPicListTemp1[cIdx]
```

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden establecer la variable NumPocTotalCurr igual a NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr+ NumPocIvCurr. En este ejemplo, RePIcSetIvCurr es el subconjunto RPS de las imágenes de referencia entre vistas.

Alternativamente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar los encabezados de segmento de acuerdo con la Tabla 4 más abajo:

Tabla 4

slice_header() {	**Descriptor**
first_slice_in_pic_flag	u(1)
...	
if(slice_header_extension_present_flag) { // siempre debe ser verdadero en MV-HEVC	
slice_header_extension_length	ue(v)
if(slice_type != I_SLICE) {	
inter_view_ref_pos_default_flag	u(1)
if(! inter_view_ref_pos_default_flag)	
inter_view__ref_start_position	ue(v)
}	
...	
}	
byte_alignment( )	
}	

En este ejemplo, se agregan dos elementos de sintaxis adicionales al encabezado de segmento después de slice_header_extensión_length, con relación a HEVC WD8. Alternativamente, los elementos de sintaxis recién introducidos se pueden presentar en otros lugares del encabezado de segmento y presentarse antes de los elementos de sintaxis de modificación de la lista de imágenes de referencia (RPLM). Inter_view_ref_pos_default_flag igual a 1 puede indicar que las imágenes de referencia entre vistas se insertan en la posición inicial predeterminada de la lista de imágenes de referencia 0 y que inter_view_ref_start_position se deriva para ser NumPocStCurrBefore. Inter_view_ref_pos_default_flag igual a 0 puede indicar que la posición inicial de las imágenes de referencia entre vistas se decide por inter_view_ref_start_position.

Alternativamente, inter_view_ref_pos_default_flag igual a 1 puede indicar que las imágenes de referencia entre vistas se insertan en la posición inicial predeterminada de la lista de imágenes de referencia 0, mientras que inter_view_ref_pos_default_flag igual a 0 puede indicar que se decide la posición inicial de las imágenes de referencia entre vistas por inter view_ref_start_position. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden derivar la posición predeterminada de la siguiente manera. Primero, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden establecer KeyPicPoc como el valor de POC de la imagen en el Búfer de Imagen Decodificada (DPB) con un valor de POC más grande entre todas las imágenes en DPB con temporalID igual a 0 y con un valor de POC menor que el POC actual. Si todas las imágenes de referencia a corto plazo tienen un valor de POC menor que KeyPicPoc, se puede derivar inter_view_ref_start_position para que sea igual a 1; de lo contrario, inter_view_ref_start_position se puede derivar para ser NumPocStCurrBefore.

En este ejemplo, inter_view_ref_start_position puede especificar la posición inicial de las imágenes de referencia entre vistas en la lista de imágenes de referencia 0 después de la inicialización de la lista de imágenes de referencia. Inter_view_ref_start_position puede tener un valor en el intervalo de 0 a min (num_ref_idx_l0_active_minus1, NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr), inclusive. Alternativamente, inter_view_ref_start_position puede tener un valor en el intervalo de 0 a num_ref_idx_10_active_minus1, inclusive. El procedimiento de codificación para la inicialización de la lista de imágenes de referencia puede ser el mismo que cualquiera de las técnicas como se describió anteriormente.

Alternativamente, los dos elementos de sintaxis inter_view_ref_pos_default_flag e inter_view_ref_start_position se pueden reemplazar por un solo elemento de sintaxis inter_view_ref_start_position_plus1. Inter_view_ref_start_position_plus1 igual a cero puede indicar que se señala una posición de inicio no explícita para las imágenes de referencia entre vistas. Para otros valores de inter_view_ref_start_position_plus1, inter_view_ref_start_position_plus 1minus 1 puede indicar la posición de inicio de las imágenes de referencia entre vistas en la lista de imágenes de referencia.

Alternativamente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar encabezados de segmento de acuerdo con la Tabla 5 más abajo, donde el texto subrayado representa adiciones con relación a HEVC WD8:

Tabla 5

slice_header() {	Descriptor
first_slice_in_pic_flag	u(1)
...	
if(slice_header_extension_present_flag) { // siempre debe ser verdadero en MV-HEVC	
slice_header_extension_length	ue(v)
if(slice_tvpe != I && viewIdx > 0)	
inter view ref start position L0 plus1	ue(v)
if(slice tvpe = = B && viewIdx > 0)	
inter view ref start position L1 plus1	ue(v)
...	
}	
byte_alignment( )	
}	

Alternativamente, los elementos de sintaxis recién introducidos (como subrayados) se pueden presentar en otros lugares del encabezado de segmento y presentarse antes de los elementos de sintaxis de RPLM. Alternativamente, la posición de inicio predeterminada para las imágenes de referencia entre vistas sólo se puede señalar para RefPicListO, y el procedimiento de decodificación para RefPicList1 puede seguir siendo el mismo que la especificación MV-HEVC actual. En este caso, el procedimiento de inicialización de la lista de imágenes de referencia para RefPicList1 puede mantenerse sin cambios.

En el ejemplo de la Tabla 5, inter_view_ref_start_position_LX_plus1 especifica la posición inicial de las imágenes de referencia entre vistas en la lista de imágenes de referencia X después de la inicialización de la lista de imágenes de referencia. Inter_view_ref_start_position_LX_plus1 está en el intervalo de 0 a min (num_ref_idx_lX_active_minus1 + 1, NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr), inclusive. Cuando inter_view_ref_start_position_LX_plus1 es igual a cero, en el ejemplo de la Tabla 5, las imágenes de referencia entre vistas están presentes en la posición predeterminada en la lista de imágenes de referencia. Para otros valores distintos de cero, en el ejemplo de la Tabla 5, inter_view_ref_start_position_LX_plus1minus 1 denota la posición de inicio de las imágenes de referencia entre vistas en la lista de imágenes de referencia inicial. Cuando no está presente, se puede inferir que inter_view_ref_start_position_LX_plus1 es un valor predeterminado, que puede ser igual a NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr + 1.

El codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para derivar un valor InterViewRefStartPosLX de la siguiente manera:

if(inter_view_ref_start_position_LX_plus1 = = 0)

InterViewRefStartPosLX = NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr

si no

InterViewRefStartPosLX = inter_view_ref_start_position_LX_plus1 – 1

Alternativamente, inter_view_ref_start_position_LX_plus1 se puede señalar como dos elementos de sintaxis, inter_view_ref_start_position_flag_LX e inter_view_ref_start_position_LX, donde inter_view_ref_start_position_LX se puede señalar como condicionado si y solo si el valor de inter_view_ref_start_position_flag_LX es igual a 1. Inter_view_ref_start_position _LX puede especificar la posición inicial de las imágenes de referencia entre vistas en la lista de imágenes de referencia X después de la inicialización de la lista de imágenes de referencia. Inter_view_ref_start_position puede estar en el intervalo de 0 a min (num_ref_idx_lX_active_minus1,

NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr), inclusive. Cuando no está presente, se puede inferir que inter_view_ref_start_position_LX es igual a (NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr).

En algunos ejemplos, cuando inter_view_ref_start_position_LX_plus1 es igual a 1, la inicialización de la lista de imágenes de referencia puede no cambiarse y por lo tanto, puede ser la misma que la especificada en el borrador de trabajo MV-HEVC.

El decodificador de vídeo 30 se puede configurar para realizar lo siguiente durante el procedimiento de decodificación, cuando se implemente de acuerdo con la Tabla 5 anterior. En particular, el decodificador de vídeo 30 puede realizar el siguiente procedimiento al decodificar un encabezado de segmento P o B. El decodificador de vídeo 30 puede establecer la variable NumRpsCurrTempListO igual a Max(num_ref_idx_10_active_minus1 + 1, NumPocTotalCurr). El decodificador de vídeo 30 puede entonces construir la lista RefPicListTempO de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList0 - NumPocIvCurr  ) {
    for( i = 0; i < NumPocStCurrBefore && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
            RefPicListTemp0[ cIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0;  i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )                                              (F-25)
            RefPicListTemp0[ cIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList0;
        cIdx++, i++ )
            RefPicListTemp0[ cIdx ] = RefPicSetLtCurr[ i ]
}
for(cIdx = NumRpsCurrTempList0-1; cIdx >= InterViewRefStartPosL0 +
    NumPocIvCurr; cdx --)
    RefPicListTemp0[ cIdx ] = RefPicListTemp0[ cIdx- NumPocIvCurr ]
for (i=0; i< NumPocIvCurr; i++)
    RefPicListTemp0[InterViewRefStartPosL0 + i] = RefPicSetIvCurr[ i ]
```

El decodificador de vídeo 30, de acuerdo con el ejemplo de la Tabla 5, puede entonces construir la lista RefPicListO de la siguiente manera:

```
for( cIdx = 0; cIdX ≤ num_ref_idx_l0_active_minus1; cIdx++)          (F-26)
    RefPicList0[ cIdx ] = ref_pic_list_modification_flag_l0 ?
        RefPicListTemp0[list_entry_l0[cIdx]] : RefPicListTemp0[cIdx]
```

Cuando el segmento es un segmento B, el decodificador de vídeo 30, de acuerdo con el ejemplo de la Tabla 5, puede establecer la variable NumRpsCurrTempList1 igual a Max(num_ref_idx_11_active_minusl + 1, NumPocTotalCurr). El decodificador de vídeo 30 puede entonces construir la lista RefPicListTemp1 de la siguiente manera:

```
cIdx = 0
while( cIdx < NumRpsCurrTempList1 – NumPocIvCurr ) {
        for( i = 0; i < NumPocStCurrAfter && cIdx < NumRpsCurrTempList1;
                cIdx++, i++ )
                        RefPicListTemp1[ cIdx ] = RefPicSetStCurrAfter[ i ]
        for( i = 0;  i < NumPocStCurrBefore && cIdx < NumRpsCurrTempList1;
                cIdx++, i++ )                                              (F-27)
                RefPicListTemp1[ cIdx ] = RefPicSetStCurrBefore[ i ]
        for( i = 0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList1;
                cIdx++, i++ )
                RefPicListTemp1[ cIdx ] = RefPicSetLtCurr[ i ]
}
for(cIdx = NumRpsCurrTempList1   1; cIdx >= InterViewRefStartPosL1 +
        NumPocIvCurr; cdx --)
        RefPicListTemp1[ cIdx ] = RefPicListTemp1[ cIdx- NumPocIvCurr ]
for (i=0; i< NumPocIvCurr; i++)
        RefPicListTemp1[InterViewRefStartPosL1 +i] = RefPicSetIvCurr[ i ]
```

Cuando el segmento es un segmento B, el decodificador de vídeo 30, de acuerdo con el ejemplo de la Tabla 5, puede construir la lista RefPicList1 de la siguiente manera:

```
for( cIdx = 0; cIdX ≤ num_ref_idx_l1_active_minus1; cIdx++)              (F-28)
        RefPicList1[ cIdx ] = ref_pic_list_modification_flag_l1 ?
                RefPicListTemp1[list_entry_l1[cIdx]] : RefPicListTemp1[cIdx]
```

De manera similar a los ejemplos descritos anteriormente, con las posiciones de inicio de referencia entre vistas señaladas para RefPIcListX se señalan en u(v), mientras que el intervalo de inter_view_ref_start_position o inter_view_ref_start_position_LX_plus1 están en un intervalo más pequeño, por lo que consume menos bits.

Por ejemplo, inter_view_ref_start_position puede estar en el intervalo de 0 a min (num_ref_idx_10_active_minus1, NumPocStCurrBefore), inclusive.

Alternativamente, como en otro ejemplo, inter_view_ref_start_position está en el intervalo de 0 a min (num_ref_idx_10_active_minus1, NumPocStCurrBefore + NumPocStCurrAfter), inclusive; y inter_view_ref_start_position_LX_plus1 está en el intervalo de min (num_ref_idx_1X_active_minus1 + 1, (X ? NumPocStCurrAfter: NumPocStCurrBefore)).

Alternativamente, como en otro ejemplo, inter_view_ref_start_position_LX_plus1 está en el intervalo de min (num_ref_idx_lX_active_minus1 + 1, NumPocStCurrBefore + NumPocStCurrAfter),

El valor predeterminado de cada uno de los elementos de sintaxis anteriores, cuando no está presente, se adapta para alinearse con el intervalo del elemento de sintaxis.

Otro ejemplo, similar a algunos otros ejemplos anteriores, inter_view_ref_pos_default_flag se extiende tanto a RefPicListO como a RefPicList1. Inter_view_ref_start_position o inter_view_ref_start_position_LX se señalan como u(v), con el siguiente intervalo. En uno de estos ejemplos, inter_view_ref_start_position está en el intervalo de 0 a min (num_ref_idx_10_active_minus1, NumPocStCurrBefore), inclusive. inter_view_ref_start_position_LX está en el intervalo de min (num_ref_idx_lX_active_minus1, (X? NumPocStCurrAfter: NumPocStCurrBefore)-1).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos de codificador o decodificador adecuados, según corresponda, tal como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados para aplicaciones específicas (ASIC), matrices de puertas lógicas programables en campo (FPGA), circuitos lógicos discretos, software, hardware, firmware o cualquier combinación de los mismos. Cada uno del codificador de vídeo 20 y decodificador de vídeo 30

se puede incluir en uno o más codificadores o decodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/decodificador combinado (CODEC). Un dispositivo que incluye un codificador de vídeo 20 y/o un decodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador, y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

La Figura 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar técnicas para construir listas de imágenes de referencia. El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de los bloques de vídeo dentro de los segmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo determinado. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) puede referirse a cualquiera de varios modos de codificación que se basan en el espacio. Los intermodos, tales como la predicción unidireccional (modo P) o bipredicción (modo B), pueden referirse a cualquiera de varios modos de codificación que se basan en el tiempo.

Como se muestra en la Figura 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama de vídeo para codificarse. En el ejemplo de la Figura 2, el codificador de vídeo 20 incluye la unidad de selección de modo 40, la memoria de imagen de referencia 64, el sumador 50, la unidad de procesamiento de transformación 52, la unidad de cuantificación 54, y la unidad de codificación de entropía 56. La unidad de selección de modo 40, a su vez, incluye la unidad de compensación de movimiento 44, la unidad de estimación de movimiento 42, la unidad de intrapredicción 46, y la unidad de partición 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye la unidad de cuantificación inversa 58, la unidad de transformación inversa 60, y el sumador 62. Puede incluirse además un filtro de desbloqueo (no se muestra en la Figura 2) para filtrar los límites de los bloques para eliminar los artefactos de bloqueo del vídeo reconstruido. Si se desea, el filtro de desbloqueo típicamente filtraría la salida del sumador 62. Pueden además, usarse filtros adicionales (en bucle o bucle posterior) además del filtro de desbloqueo. Tales filtros no se muestran por brevedad, pero si se desea, pueden filtrar la salida del sumador 50 (como un filtro en bucle).

Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o segmento de vídeo a codificarse. El segmento o trama puede dividirse en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan la codificación interpredictiva del bloque de vídeo que se recibe en relación con uno o más bloques en uno o más tramas de referencia para proporcionar predicción temporal. La unidad de procesamiento de intrapredicción 46 puede realizar alternativamente una codificación intrapredictiva del bloque de vídeo que se recibe con relación a uno o más bloques vecinos en la misma trama o segmento como el bloque a codificarse para proporcionar predicción espacial. El codificador de video 20 puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de video.

Además, la unidad de partición 48 puede particionar bloques de datos de vídeo en subbloques, en base a la evaluación de esquemas de partición previos en pasadas de codificación previas. Por ejemplo, la unidad de partición 48 puede particionar inicialmente una trama o segmento en LCU y particionar cada una de las LCU en sub-CU en base al análisis de la tasa de distorsión (por ejemplo, optimización de la tasa distorsión). La unidad de selección de modo 40 puede producir, además, una estructura de datos de árboles cuádruples indicativa de la partición de una LCU en sub-CU. Las CU de los nodos de la hoja del árbol cuádruple pueden incluir una o más PU y una o más TU.

La unidad de selección de modo 40 puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, en base a resultados de error, y proporciona el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como una trama de referencia. La unidad de selección de modo 40 proporciona, además, elementos de sintaxis, tales como vectores de movimiento, indicadores intramodo, información de partición y otra información de sintaxis similar, a la unidad de codificación de entropía 56.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar muy integradas, pero se ilustran por separado para propósitos conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el procedimiento de generar los vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama de vídeo actual o una imagen con relación a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada) con relación al bloque actual que se codifica dentro de la trama actual (u otra unidad codificada)

Un bloque predictivo es un bloque que coincide estrechamente 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 sub-enteros de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. 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 píxeles fraccionarios de la imagen de

referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con relación a las posiciones de píxeles completos y a las posiciones de píxeles fraccionarios y generar un vector de movimiento con precisión de píxeles fraccionarios.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un segmento intercodificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar de una primera lista de imágenes de referencia (List 0) o de una segunda lista de imágenes de referencia (List 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64.

De acuerdo con las técnicas de esta divulgación, cuando se codifica un segmento de una capa no base (que puede incluir un segmento de una vista no base), la unidad de estimación de movimiento 42 también puede calcular un vector de movimiento entre vistas (por ejemplo, un vector de movimiento de disparidad para la predicción entre vistas). Por lo tanto, el codificador de vídeo 20 puede añadir identificadores para imágenes de referencia entre capas (o entre vistas) a la lista de imágenes de referencia en posiciones particulares, de manera que el codificador de vídeo 20 puede codificar elementos de sintaxis representativos de las posiciones de los identificadores.

Como se analizó anteriormente, los elementos de sintaxis pueden tener valores representativos de dónde colocar las imágenes de referencia entre capas durante un procedimiento de inicialización de la imagen de referencia. Por ejemplo, los elementos de sintaxis pueden indicar que las imágenes de referencia entre capas deben colocarse consecutivamente en la lista de imágenes de referencia, que las imágenes de referencia entre vistas deben colocarse al comienzo de una lista de imágenes de referencia inicial para el segmento actual, que las imágenes de referencia entre vistas deben colocarse después de las imágenes de referencia temporal con valores de recuento de orden de imágenes (POC) más pequeños que un valor de POC del segmento actual, que las imágenes de referencia entre vistas deben colocarse después de las imágenes de referencia temporal con valores de POC mayores que un valor de POC del segmento actual, o que las imágenes de referencia entre vistas deben colocarse después de las imágenes de referencia a largo plazo.

La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación de entropía 56 y a la unidad de compensación de movimiento 44. La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar buscar o generar el bloque predictivo en base al vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente, en algunos ejemplos. Al recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia.

El sumador 50 forma un bloque de vídeo residual mediante la sustracción de los valores de píxeles del bloque predictivo de los valores de píxeles del bloque de vídeo actual que se está codificando, que forma valores de diferencia de píxeles, como se analiza más abajo. En general, la unidad de estimación de movimiento 42 realiza una estimación de movimiento con relación a los componentes de luminancia, y la unidad de compensación de movimiento 44 usa vectores de movimiento calculados en base a los componentes de luminancia para ambos componentes de crominancia y componentes de luminancia. La unidad de selección de modo 40 puede, además, generar elementos de sintaxis que se asocian con los bloques de video y el segmento de video para su uso mediante el decodificador de video 30 al decodificar los bloques de video del segmento de video.

La unidad de intrapredicción 46 puede intrapredecir un bloque actual, como una alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se describió anteriormente. En particular, la unidad de intrapredicción 46 puede determinar un modo de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 46 puede codificar un bloque actual mediante el uso de varios modos de intrapredicción, por ejemplo, durante pasadas de codificación separadas, y la unidad de intrapredicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado para usar entre los modos probados.

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

Después de seleccionar un modo de intrapredicción para un bloque, la unidad de intrapredicción 46 puede proporcionar la información indicativa del modo de intrapredicción seleccionado para el bloque a la unidad de codificación de entropía 56. La unidad de codificación de entropía 56 puede codificar la información que indica el

modo de intrapredicción que se selecciona. El codificador de vídeo 20 puede incluir en los datos de configuración del flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índice en modo de intrapredicción y una pluralidad de tablas de índice en modo de intrapredicción modificadas (que se refieren además, a tablas de asignación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índice de modo de intrapredicción y una tabla de índice de modo de intrapredicción modificada para usar en cada uno de los contextos.

El codificador de vídeo 20 forma un bloque de vídeo residual mediante la sustracción de 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 sustracción. La unidad de procesamiento de transformación 52 aplica una transformación, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar, al bloque residual, lo que produce un bloque de video que comprende valores de coeficiente de transformada residual. La unidad de procesamiento de transformación 52 puede realizar otras transformadas que son conceptualmente similares a DCT. Podrían, además, usarse transformadas de ondículas, transformadas de enteros, transformadas de subbandas u otros tipos de transformadas.

En cualquier caso, la unidad de procesamiento de transformación 52 aplica la transformada al bloque residual, lo que produce un bloque de coeficientes de transformación residual. La transformación puede convertir la información residual de un dominio de valor de píxel en un dominio de transformación, tal como un dominio de frecuencia. La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantificación 54. La unidad 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 que se asocia con algunos o todos los coeficientes. El grado de cuantificación puede modificarse mediante el ajuste de un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede luego realizar una exploración de la matriz que incluye los coeficientes de transformación cuantificados. Alternativamente, la unidad de codificación de entropía 56 puede realizar la exploración.

Después de la cuantificación, la entropía de la unidad de codificación de entropía 56 codifica los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación de entropía 56 puede realizar codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto en base a sintaxis (SBAC), codificación de entropía de partición de intervalo de probabilidad (PIPE) u otra técnica de codificación de entropía. En el caso de la codificación de entropía basada en el contexto, el contexto puede basarse en bloques vecinos. Después de la codificación de entropía mediante la unidad de codificación de entropía 56, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, decodificador de vídeo 30) o archivarse para su posterior transmisión o recuperación.

La unidad de cuantificación inversa 58 y la unidad de transformación inversa 60 aplican la cuantificación inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso posterior como bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia al añadir el bloque residual a un bloque predictivo de una de las tramas de la memoria de imágenes de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles subenteros para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento producido por la unidad de compensación de movimiento 44 para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria de imagen de referencia 64. El bloque de video reconstruido puede usarse mediante la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para intercodificar un bloque en una trama de video subsiguiente.

Aunque se describe principalmente con respecto a la predicción temporal, el codificador de vídeo 20 también se puede configurar para realizar predicciones entre vistas. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 también se pueden configurar para realizar una estimación de disparidad y una compensación de disparidad, respectivamente, por ejemplo, con relación a una imagen de referencia de predicción entre vistas. Por lo tanto, el codificador de vídeo 20 se puede configurar para utilizar las técnicas de esta divulgación para construir una lista de imágenes de referencia que incluye los identificadores para imágenes de referencia entre vistas.

De acuerdo con ciertas técnicas de esta divulgación, el codificador de vídeo 20 se puede configurar para codificar los datos de vídeo de múltiples capas (que pueden incluir datos de vídeo multivista). Por ejemplo, el codificador de vídeo 20 se puede configurar para codificar una o más capas (por ejemplo, una o más vistas) de acuerdo con una extensión de un estándar de codificación de vídeo base. Como ejemplo, el codificador de vídeo 20 se puede configurar para codificar las capas no base de acuerdo con una extensión S-HEVC, o las vistas no base de acuerdo con una extensión MV-HEVC o 3D-HEVC. El codificador de vídeo 20 se puede configurar para codificar los datos de vídeo de la capa base de acuerdo con el estándar de codificación de vídeo base, por ejemplo, HEVC.

Al codificar los datos de la capa base (o vista base) de acuerdo con un estándar de codificación de vídeo base, el codificador de vídeo 20 no necesita codificar los elementos de sintaxis para una extensión del estándar de

codificación de vídeo base para la capa base. Por ejemplo, con respecto a los ejemplos anteriores, debido a que los datos de vídeo de la capa base no se pueden predecir entre capas, no habría razón para codificar los elementos de sintaxis representativos de dónde colocar los identificadores para imágenes de referencia entre capas en una lista de imágenes de referencia para la capa base. Por lo tanto, el codificador de vídeo 20 puede omitir la codificación de dichos elementos de sintaxis para los datos de vídeo de la capa base.

Sin embargo, el codificador de vídeo 20 puede codificar datos de vídeo que no son de la capa base (por ejemplo, segmentos que tienen un identificador de capa en un encabezado de segmento con un valor distinto de cero) de acuerdo con una extensión del estándar de codificación de vídeo base, tal como S-HEVC, MV-HEVC, o 3D-HEVC. Por lo tanto, el codificador de vídeo 20 puede codificar tanto los elementos de sintaxis para el estándar de codificación de vídeo base como otro conjunto adicional de elementos de sintaxis que se definen por la extensión. Tal conjunto adicional de elementos de sintaxis puede incluir, por ejemplo, uno o más elementos de sintaxis representativos de dónde colocar las imágenes de referencia entre capas en una lista de imágenes de referencia, como se analizó anteriormente.

El codificador de vídeo 20 de la Figura 2 representa un ejemplo de un codificador de vídeo que se configura para codificar un valor para un identificador de capa en un encabezado de segmento para un segmento actual en una capa actual de datos de vídeo de múltiples capas y, cuando el valor para el identificador de capa no es igual a cero, codificar un primer conjunto de elementos de sintaxis de acuerdo con un estándar de codificación de vídeo base, y codificar un segundo conjunto de uno o más elementos de sintaxis de acuerdo con una extensión del estándar de codificación de vídeo base.

El codificador de vídeo 20 de la Figura 2 también representa un ejemplo de un codificador de vídeo que se configura para determinar, para la construcción de una lista de imágenes de referencia para un segmento actual de una vista actual, una posición para un identificador de una imagen de referencia entre vistas de una vista de referencia, construir la lista de imágenes de referencia de manera que el identificador de la imagen de referencia entre vistas se coloque en la posición determinada de la lista de imágenes de referencia, codifique un valor de índice correspondiente a la posición determinada para una porción del segmento actual, y codifique la porción del segmento actual mediante el uso de la imagen de referencia entre vistas en base al valor del índice.

La Figura 3 es un diagrama de bloques que ilustra un ejemplo del decodificador de vídeo 30 que puede implementar técnicas para construir listas de imágenes de referencia. En el ejemplo de la Figura 3, el decodificador de vídeo 30 incluye una unidad de decodificación de entropía 70, una unidad de compensación de movimiento 72, una unidad de intrapredicción 74, una unidad de cuantificación inversa 76, una unidad de transformación inversa 78, una memoria de imagen de referencia 82 y un sumador 80. El decodificador de vídeo 30 puede, en algunos ejemplos, realizar una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 (Figura 2).

De acuerdo con las técnicas de esta divulgación, el decodificador de vídeo 30 puede determinar si un segmento de datos de vídeo multicapa está en una capa base o en una capa no base (por ejemplo, una capa de mejora o una vista dependiente). Por ejemplo, el decodificador de vídeo 30 puede decodificar un identificador de capa en un encabezado de segmento del segmento que tiene un valor que indica que el segmento está en una capa base o en una capa no base. Para los datos de vídeo multivista, el identificador de capa puede corresponder a un identificador de vista (view id). En algunos ejemplos, un valor de cero para el identificador de capa indica que el segmento está en la capa base, y un valor distinto de cero para el identificador de capa indica que el segmento está en una capa no base, por ejemplo, una capa de mejora o una vista dependiente.

Cuando el decodificador de vídeo 30 determina que el segmento está en la capa base, el decodificador de vídeo 30 puede decodificar un primer conjunto de elementos de sintaxis, conforme a un estándar de codificación de vídeo base. Cuando el decodificador de vídeo 30 determina que el segmento está en una capa no base (por ejemplo, una capa de mejora o una vista dependiente), el decodificador de vídeo 30 puede decodificar un segundo conjunto de elementos de sintaxis, conforme a una extensión del estándar de codificación de vídeo base, además del primer conjunto de elementos de sintaxis. El primer y el segundo conjunto de elementos de sintaxis pueden codificarse como parte de cualquiera de, o cualquier combinación de, un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un encabezado de segmento, o similar.

En general, el segundo conjunto de elementos de sintaxis incluye datos para la extensión del estándar de codificación de vídeo base que no son relevantes para el estándar de codificación de vídeo base. Por ejemplo, el segundo conjunto de elementos de sintaxis puede incluir uno o más elementos de sintaxis relacionados con cómo construir una lista de imágenes de referencia para incluir las imágenes de referencia entre capas (que incluye entre vistas). Por ejemplo, uno o más elementos de sintaxis, en este ejemplo, pueden indicar que las imágenes de referencia entre vistas deben colocarse consecutivamente en la lista de imágenes de referencia, colocadas al comienzo de una lista de imágenes de referencia inicial, después de las imágenes de referencia temporal con valores de recuento de orden de imágenes (POC) más pequeños que un valor de POC de un segmento actual (para el que se construye la lista de imágenes de referencia), o después de las imágenes de referencia a largo plazo.

En el ejemplo analizado anteriormente, el decodificador de vídeo 30 puede construir una lista de imágenes de referencia, para segmentos en una capa de mejora o vista dependiente, de acuerdo con los elementos de sintaxis adicionales. El procedimiento de construcción de la lista de imágenes de referencia puede incluir la construcción de una lista de imágenes de referencia temporal que incluye imágenes de referencia temporales en un orden que se basa, al menos en parte, en la posición de las imágenes de referencia entre capas o entre vistas.

Durante el procedimiento de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa los bloques de vídeo de un segmento de vídeo codificado y los elementos de sintaxis asociados del codificador de vídeo 20. La unidad de decodificación de entropía 70 del decodificador de video 30 decodifica la entropía del flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción, y otros elementos de sintaxis. La unidad de decodificación de entropía 70 envía los vectores de movimiento y otros elementos de sintaxis a la unidad de compensación de movimiento 72. El decodificador de vídeo 30 puede recibir los elementos de sintaxis a nivel de segmento de vídeo y/o a nivel de bloque de vídeo.

La unidad de compensación de movimiento 72 puede generar datos de predicción en base a vectores de movimiento recibidos desde la unidad de decodificación de entropía 70, mientras que la unidad de intrapredicción 74 puede generar datos de predicción en base a los indicadores de modo de intrapredicción recibidos desde la unidad de decodificación de entropía 70. Cuando el segmento de vídeo se codifica como un segmento intracodificado (I), la unidad de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del segmento de vídeo actual en base a un modo de intrapredicción señalizado y los datos de bloques previamente decodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un segmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 produce bloques predictivos para un bloque de vídeo del segmento de vídeo actual en base a los vectores de movimiento y otros elementos de sintaxis recibidos de la entropía unidad de decodificación 70. Los bloques predictivos se pueden producir 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 tramas de referencia, Lista 0 y Lista 1, mediante el uso de técnicas de construcción por defecto en base a las imágenes de referencia almacenadas en la memoria de imágenes de referencia 82 (también denominada memoria intermedia de imágenes decodificadas).

La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del segmento de vídeo actual al analizar los vectores de movimiento y otros elementos de sintaxis, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se decodifica. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra o interpredicción) usado para codificar los bloques de vídeo del segmento de vídeo, un tipo de segmento de interpredicción (por ejemplo, segmento B, segmento P, o segmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de video intercodificado del segmento, estado de interpredicción para cada bloque de vídeo intercodificado del segmento, y otra información para decodificar los bloques de vídeo en el segmento de vídeo actual.

La unidad de compensación de movimiento 72 puede, además, realizar una interpolación en base a filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como se usan mediante el codificador de video 20 durante la codificación de los bloques de video para calcular valores interpolados para píxeles subenteros de bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación que se usan mediante el codificador de video 20 a partir de los elementos de sintaxis que se reciben y usar los filtros de interpolación para producir bloques predictivos.

La unidad de cuantificación inversa 76 cuantifica inversamente, es decir, descuantifica, los coeficientes de transformación cuantificados que se proporcionan en el flujo de bits y decodificados mediante la unidad de decodificación de entropía 70. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QP_Y que se calcula mediante el decodificador de video 30 para cada bloque de video en el segmento de video para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse.

La unidad de transformación inversa 78 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada de entero inverso o un procedimiento de transformada inversa conceptualmente similar, a los coeficientes de transformación para producir bloques residuales en el dominio de píxeles.

Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual en base a los vectores de movimiento y otros elementos de sintaxis, el decodificador de vídeo 30 forma un bloque de vídeo decodificado que suma los bloques residuales de la unidad de transformación inversa 78 con los bloques predictivos correspondientes generado por la unidad de compensación de movimiento 72. El sumador 80 representa el componente o componentes que realizan esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques decodificados con el fin de eliminar los artefactos de bloqueo. Pueden además, usarse 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 de cualquier otro modo la calidad del vídeo. Los

bloques de vídeo decodificados en una trama o imagen dadas se almacenan luego en la memoria de imágenes de referencia 82, que almacena las imágenes de referencia usadas para la compensación de movimiento subsiguiente. La memoria de imágenes de referencia 82 también almacena el vídeo decodificado para una presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la Figura 1.

Aunque se describe principalmente con respecto a la predicción temporal, el decodificador de vídeo 30 también se puede configurar para realizar predicciones entre vistas. La unidad de compensación de movimiento 72 también se puede configurar para realizar la compensación de disparidad, por ejemplo, con relación a una imagen de referencia de predicción entre vistas. Por lo tanto, el decodificador de vídeo 30 se puede configurar para utilizar las técnicas de esta divulgación para construir una lista de imágenes de referencia que incluye los identificadores para imágenes de referencia entre vistas.

El decodificador de vídeo 30 de la Figura 3 representa un ejemplo de un codificador de vídeo que se configura para codificar un valor para un identificador de capa en un encabezado de segmento para un segmento actual en una capa actual de datos de vídeo de múltiples capas y, cuando el valor para el identificador de capa no es igual a cero, codificar un primer conjunto de elementos de sintaxis de acuerdo con un estándar de codificación de vídeo base, y codificar un segundo conjunto de uno o más elementos de sintaxis de acuerdo con una extensión del estándar de codificación de vídeo base.

El decodificador de vídeo 30 de la Figura 3 también representa un ejemplo de un codificador de vídeo que se configura para determinar, para la construcción de una lista de imágenes de referencia para un segmento actual de una vista actual, una posición para un identificador de una imagen de referencia entre vistas de una vista de referencia, construir la lista de imágenes de referencia de manera que el identificador de la imagen de referencia entre vistas se coloque en la posición determinada de la lista de imágenes de referencia, codifique un valor de índice correspondiente a la posición determinada para una porción del segmento actual, y codifique la porción del segmento actual mediante el uso de la imagen de referencia entre vistas en base al valor del índice.

La Figura 4 es un diagrama de flujo que ilustra un procedimiento de ejemplo para construir una lista de imágenes de referencia y usar la lista de imágenes de referencia cuando se codifica una porción de un segmento actual. El codificador de vídeo 20 y el decodificador de vídeo 30 se pueden configurar para realizar el procedimiento de la Figura 4, o un procedimiento sustancialmente similar. Para fines de explicación, el procedimiento de la Figura 4 se explica con respecto al decodificador de vídeo 30, aunque debe entenderse que el codificador de vídeo 20 también se puede configurar para realizar un procedimiento similar. El codificador de vídeo 20 puede realizar determinadas etapas adicionales o alternativas, como se indica más abajo.

El decodificador de vídeo 30 puede primero recibir y decodificar una imagen de referencia entre vistas (150). En particular, el decodificador de vídeo 30 puede decodificar imágenes (que incluye uno o más segmentos) en varias vistas, por ejemplo, en un primer orden temporal. Es decir, el flujo de bits puede construirse de tal manera que los datos para todas las imágenes en la misma instancia temporal estén presentes juntos en el flujo de bits, por ejemplo, en una unidad de acceso común. Por lo tanto, el decodificador de vídeo 30 puede decodificar las imágenes de una unidad de acceso en el orden en que aparecen estas imágenes en la unidad de acceso y, por tanto, en el flujo de bits. Cuando el procedimiento se realiza por el codificador de vídeo 20, el codificador de vídeo 20, u otra unidad acoplada comunicativamente al codificador de vídeo 20, tal como un multiplexor, puede ensamblar los datos de vídeo en el flujo de bits en esta disposición, por ejemplo, al encapsular los datos de vídeo para imágenes de una instancia temporal común en una unidad de acceso. El decodificador de vídeo 30 puede almacenar en memoria intermedia los datos decodificados para la imagen de referencia entre vistas en la memoria de imagen de referencia 82.

El decodificador de vídeo 30 puede entonces obtener los datos de vídeo para un segmento actual de una imagen (componente de vista) en una vista diferente de la misma unidad de acceso. Los datos del segmento actual pueden codificarse con relación a la imagen de referencia entre vistas, mediante el uso de la predicción entre vistas. Cuando el procedimiento se realiza por el codificador de vídeo 20, el codificador de vídeo 20 puede probar varios modos de predicción para determinar si la predicción entre vistas es apropiada, por ejemplo, mediante el uso de la optimización de distorsión de velocidad. En cualquier caso, el decodificador de vídeo 30 puede determinar una posición para un identificador de imagen de referencia entre vistas en una lista de imágenes de referencia (152).

La lista de imágenes de referencia puede incluir un conjunto ordenado de identificadores para imágenes de referencia, por ejemplo, almacenados en la memoria de imágenes de referencia 82 (también denominada memoria intermedia de imágenes decodificadas). La lista de imágenes de referencia, de acuerdo con las técnicas de esta divulgación, puede incluir identificadores para imágenes de referencia temporales (imágenes de referencia de la vista actual) y uno o más identificadores para imágenes de referencia entre vistas. El decodificador de vídeo 30 se puede configurar para determinar la posición del identificador de imagen de referencia entre vistas mediante el uso de cualquiera de las técnicas descritas anteriormente, solo o en cualquier combinación. El decodificador de vídeo 30 puede entonces construir la lista de imágenes de referencia (154), de manera que el identificador de imágenes de referencia entre vistas se coloque en la posición determinada en la etapa 152.

El decodificador de vídeo 30 puede codificar entonces un valor de índice del identificador de imagen de referencia entre vistas en la lista de imágenes de referencia para el segmento actual (156). Es decir, en este ejemplo, el decodificador de vídeo 30 puede recibir el valor del índice y decodificar el valor del índice, donde en este caso, el valor del índice corresponde a la posición del identificador de la imagen de referencia entre vistas en la lista de imágenes de referencia. De esta manera, el decodificador de vídeo 30 puede determinar que la imagen de referencia entre vistas se va a usar para decodificar una porción del segmento actual. El valor de índice puede codificarse como información de movimiento para un bloque, tal como una unidad de predicción (PU), del segmento actual y, en particular, como índice de referencia en la información de movimiento.

Cuando el procedimiento se realiza por el codificador de vídeo 20, el codificador de vídeo 20 puede, como se explicó anteriormente, determinar que un bloque particular del segmento actual debe predecirse mediante el uso de la predicción entre vistas y, por lo tanto, codificar el valor del índice en consecuencia. También debe entenderse que en general, el valor del índice puede corresponder a cualquiera de los identificadores de imágenes de referencia en la lista de imágenes de referencia; en este ejemplo específico, sin embargo, el índice corresponde al identificador de la imagen de referencia entre vistas, para ilustrar las técnicas de esta divulgación.

Además, debido a que el valor del índice corresponde al identificador de la imagen de referencia entre vistas en la lista de imágenes de referencia, el decodificador de vídeo 30 puede decodificar la porción correspondiente del segmento actual mediante el uso de la imagen de referencia entre vistas (158). Por ejemplo, al asumir que la porción corresponde a un bloque, por ejemplo, una PU, el decodificador de vídeo 30 puede recibir otra información de movimiento que defina, por ejemplo, un vector de movimiento de disparidad para el bloque. El decodificador de vídeo 30 (por ejemplo, la unidad de compensación de movimiento 72 del decodificador de vídeo 30) puede usar el vector de movimiento de disparidad para recuperar un bloque predictivo para el bloque actual. El decodificador de vídeo 30 también puede decodificar la información residual (por ejemplo, coeficientes de transformación cuantificados) y combinar la información residual (después de la cuantificación inversa y la transformación inversa) con el bloque predictivo para decodificar el bloque actual. El codificador de vídeo 20, por otro lado, puede calcular la información residual como un conjunto de diferencias píxel a píxel entre un bloque original y el bloque predictivo, y luego transformar y cuantificar la información residual.

De esta manera, el procedimiento de la Figura 4 representa un ejemplo de un procedimiento de codificación (por ejemplo, codificación o decodificación) de datos de vídeo, el procedimiento que incluye determinar, para la construcción de una lista de imágenes de referencia para un segmento actual de una vista actual, una posición para una imagen de referencia entre vistas de una vista de referencia, construir la lista de imágenes de referencia de manera que el identificador de la imagen de referencia entre vistas se ubique en la posición determinada de la lista de imágenes de referencia, codificar un valor de índice correspondiente a la posición determinada para una porción del segmento actual, y codificar la porción del segmento actual mediante el uso de la imagen de referencia entre vistas en base al valor del índice.

La Figura 5 es un diagrama de flujo que ilustra un procedimiento de ejemplo para codificar datos de vídeo multicapa mediante el uso de diferentes estándares de codificación de vídeo, de acuerdo con las técnicas de esta divulgación. Aunque se explica con respecto al codificador de vídeo 20 de las Figuras 1 y 2, debe entenderse que otros dispositivos de codificación de vídeo se pueden configurar para realizar una técnica similar. Por ejemplo, un codificador de vídeo o transcodificador de vídeo se puede configurar para realizar técnicas similares a las de la Figura 5.

Inicialmente, el codificador de vídeo 20 puede codificar los datos de vídeo de una capa base (200). En particular, el codificador de vídeo 20 puede codificar los datos de vídeo de la capa base de acuerdo con un estándar de codificación de vídeo base, tal como HEVC. Por ejemplo, para las imágenes intracodificadas (I-pictures), el codificador de vídeo 20 puede codificar los bloques de uno o más segmentos mediante el uso de la intrapredicción, y para las imágenes intercodificadas (imágenes P e imágenes B), el codificador de vídeo 20 puede codificar los bloques de uno o más segmentos mediante el uso de cualquier intrapredicción, interpredicción direccional, o (para imágenes B) bipredicción. Tal como se codifican, los datos de vídeo de la capa base pueden formar un flujo de bits que se ajuste al estándar de codificación de vídeo base.

El codificador de vídeo 20 también puede codificar un identificador de capa que tiene un valor de cero en las cabeceras de segmento de los segmentos en la capa base (202). El valor de cero se usa cuando un valor de cero para un identificador de capa corresponde a la capa base. Para la codificación de vídeo multivista, se supone que el valor de cero para el identificador de capa (por ejemplo, un identificador de vista) corresponde a una vista base. En los ejemplos donde un valor distinto de cero para un identificador de capa (o vista) corresponde a la capa/vista base, el codificador de vídeo 20 puede codificar el valor para el identificador de capa en un encabezado de segmento de un segmento de la capa base que corresponde a la capa base.

Además, el codificador de vídeo 20 puede codificar un primer conjunto de elementos de sintaxis para los segmentos de la capa base (204). Por ejemplo, el codificador de vídeo 20 puede codificar elementos de sintaxis de uno o más conjuntos de parámetros de secuencia (SPS), conjuntos de parámetros de imagen (PPS) y/o encabezados de segmento para segmentos de la capa base. Estos elementos de sintaxis pueden ajustarse a las especificaciones del

estándar de codificación de vídeo base. Además, los elementos de sintaxis no necesitan incluir los datos de sintaxis de una extensión del estándar de codificación de vídeo base.

El codificador de vídeo 20 puede entonces codificar los datos de vídeo de una capa no base mediante el uso de la predicción entre capas (206). Es decir, el codificador de vídeo 20 puede codificar al menos algunos bloques de la capa no base (por ejemplo, una capa de mejora de una extensión escalable o una vista dependiente para los datos de vídeo multivista) mediante el uso de la predicción entre capas (o entre vistas), por ejemplo, con relación a la capa base. En particular, el codificador de vídeo 20 puede codificar los datos de vídeo de la capa no base mediante el uso de una extensión del estándar de codificación de vídeo base, es decir, el estándar de codificación de vídeo base al que se ajusta la capa base. Al asumir que el estándar de codificación de vídeo base es HEVC, la extensión puede corresponder a MV-HEVC, 3D-HEVC o S-HEVC, en algunos ejemplos. Por supuesto, el codificador de vídeo 20 también puede codificar ciertos bloques de la capa no base mediante el uso de la interpredicción y/o la predicción temporal intracapa. Sin embargo, a modo de ejemplo, se supone que el codificador de vídeo 20 codifica al menos algunos bloques mediante el uso de la predicción entre capas (o entre vistas).

El codificador de vídeo 20 también puede codificar un identificador de capa que tiene un valor que no es igual a cero en los segmentos de la capa no base (208). Se supone, en este ejemplo, que un valor distinto de cero para el identificador de capa corresponde a una capa no base (o vista no base, para codificación de vídeo multivista). Sin embargo, en los casos en los que una capa no base puede tener un identificador de capa de valor cero, el codificador de vídeo 20 puede codificar el identificador de capa para que tenga un valor que indique que la capa no base no es una capa base.

Además, el codificador de vídeo 20 puede codificar valores para el primer conjunto de elementos de sintaxis y un segundo conjunto de elementos de sintaxis para los segmentos de la capa no base (210). Es decir, el codificador de vídeo 20 puede codificar valores para el conjunto de elementos de sintaxis que se ajustan al estándar de codificación de vídeo base, así como también uno o más elementos de sintaxis que se ajustan a la extensión del estándar de codificación de vídeo de la capa base. Por ejemplo, el codificador de vídeo 20 puede codificar elementos de sintaxis que se ajusten a una o más de las Tablas de la 1-5, como se analizó anteriormente. Es decir, en algunos ejemplos, el codificador de vídeo 20 puede codificar elementos de sintaxis indicativos de dónde deben colocarse los identificadores para las imágenes de referencia entre capas (o entre vistas) dentro de una lista de imágenes de referencia para los segmentos de una capa no base. Uno o más elementos de sintaxis que se ajustan a la extensión del estándar de codificación de vídeo de la capa base pueden incluirse dentro de cualquiera o todos los SPS, PPS, y/o encabezados de segmento para la capa no base. El codificador de vídeo 20 puede codificar los elementos de sintaxis de acuerdo con el procedimiento de ejemplo de la Figura 4.

De esta manera, el procedimiento de la Figura 5 representa un ejemplo de un procedimiento de codificación de datos de vídeo, el procedimiento que incluye la codificación de datos de vídeo de un segmento actual de acuerdo con un estándar de codificación de vídeo base cuando el segmento actual forma parte de una capa base de datos de vídeo multicapa, y la codificación de los datos de vídeo del segmento actual de acuerdo con una extensión del estándar de codificación de vídeo base cuando el segmento actual forma parte de una capa no base de los datos de vídeo multicapa, que codifica un valor para un identificador de capa en un encabezado de segmento para el segmento actual, en el que el valor del identificador de capa indica si el segmento actual forma parte de la capa base o la capa no base y, cuando el segmento actual forma parte de la capa no base: codificar un primer conjunto de elementos de sintaxis de acuerdo con el estándar de codificación de vídeo base, y codificar un segundo conjunto de uno o más elementos de sintaxis de acuerdo con la extensión del estándar de codificación de vídeo base.

La Figura 6 es un diagrama de flujo que ilustra un procedimiento de ejemplo para decodificar los datos de vídeo de acuerdo con las técnicas de esta divulgación. Aunque se describe principalmente con respecto al decodificador de vídeo 30 de las Figuras 1 y 3, debe entenderse que el procedimiento de la Figura 6 puede realizarse mediante cualquier dispositivo de decodificación de vídeo, por ejemplo, un decodificador de vídeo o un transcodificador de vídeo.

El decodificador de vídeo 30 puede recibir un segmento de una capa, y determinar la capa del segmento en base a un valor de un identificador de capa incluido en un encabezado de segmento para el segmento. En el ejemplo de la Figura 6, el decodificador de vídeo 30 primero decodifica un identificador de capa de 0 en un segmento de la capa base (230). En particular, el decodificador de vídeo 30 determina que el segmento está en la capa base porque el identificador de capa es igual a cero, en este ejemplo. Por supuesto, si se asigna un valor diferente para el identificador de capa a la capa base, el decodificador de vídeo 30 puede determinar que un segmento está en la capa base cuando el identificador de capa señalado en el encabezado de segmento para el segmento es igual a ese valor. Además, debe entenderse que el identificador de capa puede corresponder a un identificador de una capa en la codificación multicapa, que incluye identificadores de vista para la codificación multivista.

En base a la determinación de que el segmento está en la capa base, el decodificador de vídeo 30 puede decodificar un primer conjunto de elementos de sintaxis para los segmentos de la capa base (232). Por ejemplo, el decodificador de vídeo 30 puede decodificar los elementos de sintaxis de acuerdo con un estándar de codificación de vídeo base, tal como HEVC. En algunos ejemplos, esto puede incluir la selección de un analizador que se

configura para analizar datos de acuerdo con el estándar de codificación de vídeo base. Por ejemplo, el analizador se puede configurar de acuerdo con una gramática que incluye entradas para el primer conjunto de elementos de sintaxis, sin incluir entradas para otros elementos de sintaxis (por ejemplo, elementos de sintaxis de una extensión del estándar de codificación de vídeo base). Los elementos de sintaxis pueden incluir, por ejemplo, los conjuntos de parámetros de secuencia (SPS), los conjuntos de parámetros de imagen (PPS), y/o los encabezados de segmento.

El decodificador de vídeo 30 puede decodificar entonces los datos de vídeo de la capa base (234). Por ejemplo, el decodificador de vídeo 30 puede determinar si los bloques de los segmentos de la capa base se intracodifican o intercodifican, y decodificar los bloques en consecuencia. Una vez más, a la luz de la determinación de que los datos de vídeo están en la capa base, el decodificador de vídeo 30 puede decodificar los datos de vídeo mediante el uso de herramientas de codificación del estándar de codificación de vídeo base, por ejemplo, HEVC.

El decodificador de vídeo 30 puede decodificar entonces los identificadores de capa que tienen un valor distinto de cero en un segmento de una capa no base (236). En particular, después de decodificar un valor distinto de cero para un identificador de capa, el decodificador de vídeo 30 puede determinar que la capa para el segmento no es una capa base. Por ejemplo, el decodificador de vídeo 30 puede determinar que la capa es una capa de mejora, para la codificación de vídeo escalable, o una vista dependiente, para la codificación de vídeo multivista. De nuevo, esto asume que un valor de cero corresponde a la capa base; en los casos donde un valor distinto de cero corresponde a la capa base, el decodificador de vídeo 30 puede determinar que un segmento está en una capa no base cuando un identificador de capa para el segmento tiene un valor distinto del valor correspondiente a la capa base.

En base a la determinación de que el segmento está en una capa no base, el decodificador de vídeo 30 puede decodificar el primer conjunto de elementos de sintaxis (descrito anteriormente con respecto a la capa base) y un segundo conjunto de elementos de sintaxis para los segmentos de la capa no base (238). El segundo conjunto de elementos de sintaxis puede corresponder a elementos de sintaxis que son particulares de una extensión del estándar de codificación de vídeo base. Es decir, la capa base puede codificarse de acuerdo con el estándar de codificación de vídeo base, mientras que las capas no base pueden codificarse de acuerdo con la extensión del estándar de codificación de vídeo base.

En consecuencia, el decodificador de vídeo 30 puede decodificar tanto los elementos de sintaxis que se ajustan al estándar de codificación de vídeo base (el primer conjunto de elementos de sintaxis) así como también los elementos de sintaxis que se ajustan a la extensión del estándar de codificación de vídeo base (el segundo conjunto de elementos de sintaxis). Al asumir que el estándar de codificación de vídeo base es HEVC, la extensión puede corresponder a, por ejemplo, S-HEVC, MV-HEVC, o 3D-HEVC. Como se explicó anteriormente, el decodificador de vídeo 30 puede seleccionar un analizador que se configura de acuerdo con la extensión del estándar de codificación de vídeo base, de manera que el analizador se construya de acuerdo con una gramática que incluya entradas para el primer conjunto de elementos de sintaxis y el segundo conjunto de elementos de sintaxis.

El decodificador de vídeo 30 puede decodificar entonces los datos de vídeo de la capa no base mediante el uso de al menos alguna predicción entre capas (240). El decodificador de vídeo 30 puede decodificar parte de la capa no base mediante el uso de técnicas de codificación intracapas o interpredictiva, pero se supone, en este ejemplo, que al menos algunos bloques de la capa no base se codifican mediante el uso de predicción entre capas. Por ejemplo, algunos de los bloques pueden codificarse mediante el uso de la predicción entre vistas. De esta manera, el decodificador de vídeo 30 puede usar herramientas de codificación de la extensión del estándar de codificación de vídeo base para decodificar bloques que se predicen entre capas.

De acuerdo con otras técnicas de esta divulgación, el segundo conjunto de elementos de sintaxis puede incluir datos indicativos de cómo decodificar los bloques predictivos entre capas. Por ejemplo, el segundo conjunto de elementos de sintaxis puede indicar cómo construir y/o modificar una lista de imágenes de referencia, para incluir imágenes de referencia entre capas (que pueden incluir entre vistas). Mediante el uso de estos elementos de sintaxis, el decodificador de vídeo 30 puede construir o modificar una lista de imágenes de referencia, por ejemplo, como se analizó anteriormente con respecto a la Figura 4.

De esta manera, el procedimiento de la Figura 6 es un ejemplo de un procedimiento para decodificar los datos de vídeo, el procedimiento que incluye decodificar un valor para un identificador de capa en un encabezado de segmento para un segmento actual en una capa actual de datos de vídeo multicapa y, cuando el valor para el identificador de capa no es igual a cero: decodificar un primer conjunto de elementos de sintaxis de acuerdo con un estándar de codificación de vídeo base, y decodificar un segundo conjunto de uno o más elementos de sintaxis de acuerdo con una extensión del estándar de codificación de vídeo base.

Debe reconocerse que, en función del ejemplo, ciertos actos o eventos de cualquiera de las técnicas que se describen en la presente memoria pueden realizarse en una secuencia diferente, pueden adicionarse, fusionarse o omitirse por completo (por ejemplo, no todos los actos o eventos que se describen son necesario para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse concurrentemente, por ejemplo, a través de procesamiento de multihilos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, microprograma o cualquiera de sus combinaciones. Si se implementan en software, las funciones pueden almacenarse en o transmitirse como una o más instrucciones o código en un medio legible por ordenador y que se ejecutan mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible, tal como medios de almacenamiento de datos, o medios de comunicación, que incluyen cualquier medio que facilite transferir un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medios de almacenamiento legibles por ordenador tangibles que no son transitorios o (2) un medio de comunicación como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder por 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.

A manera de ejemplo, y no de limitación, tales medios legibles por ordenador pueden comprender RAM, ROM, EPROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnéticos, memoria flash o cualquier otro medio que pueda usarse para almacenar el código del programa deseado en forma de instrucciones o estructuras de datos y al que puede accederse por un ordenador. También, cualquier conexión se califica apropiadamente como un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor, u otra fuente remota mediante el uso de un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de suscriptor digital (DSL), o las tecnologías inalámbricas como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, pero en cambio se dirigen a medios de almacenamiento tangibles y no transitorios. Disco y disco, como se usa en la presente memoria, incluye el disco compacto (CD), el disco de láser, el disco óptico, el disco digital versátil (DVD), el disquete, y el disco Blu-ray donde existen discos que usualmente reproducen los datos de manera magnética, mientras que otros discos reproducen los datos de manera óptica con láseres. Las combinaciones de los medios anteriores se pueden incluir también dentro del ámbito de los medios legibles por ordenador.

Las instrucciones pueden ejecutarse por uno o más procesadores, como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados para aplicaciones específicas (ASIC), matriz de puertas lógicas programable en campo (FPGA) u otro circuito integrado lógico o discreto equivalente. En consecuencia, el término "procesador", como se usa en la presente memoria puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en la presente memoria. Además, en algunos aspectos, la funcionalidad descrita en la presente memoria puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y decodificación, o incorporarse en un códec combinado. Asimismo, las técnicas se podrían implementar completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen varios componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización por diferentes unidades de hardware. En lugar de, como se describió anteriormente, varias unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describió anteriormente, junto con software y/o microprograma adecuados.

REIVINDICACIONES

1. Un procedimiento para decodificar los datos de vídeo multivista de acuerdo con un estándar de codificación de vídeo de alta eficiencia, HEVC, comprendiendo el procedimiento:

decodificar una pluralidad de elementos de sintaxis en un encabezado de segmento de un segmento actual en una vista actual de los datos de vídeo multivista de acuerdo con una especificación base de HEVC;
decodificar un valor para un identificador de capa en el encabezado de segmento del segmento actual;
determinar si el valor para el identificador de capa es igual a cero (230, 236), en el que un valor para el identificador de capa que es igual a cero indica que el segmento actual forma parte de una vista base, y en el que un valor para el identificador de capa que no es igual a cero indica que el segmento actual forma parte de una vista no base; y
solo cuando se determina que el valor del identificador de capa no es igual a cero:
decodificar uno o más elementos de sintaxis adicionales en el encabezado de segmento del segmento actual de acuerdo con una extensión de codificación de vídeo multivista (MV-HEVC) a la especificación de base de HEVC (238), en el que uno o más elementos de sintaxis adicionales incluyen un elemento de sintaxis representativo de si las imágenes de referencia entre vistas se colocan al principio de una lista de imágenes de referencia inicial del segmento actual, inmediatamente después de las imágenes de referencia temporal con valores de recuento de orden de imágenes (POC) más pequeños que un valor de POC actual para la lista de imágenes de referencia inicial, inmediatamente después de las imágenes de referencia temporales con valores de recuento de orden de imágenes (POC) mayores que un valor de POC actual para la lista de imágenes de referencia inicial, o inmediatamente después de las imágenes de referencia a largo plazo en la lista de imágenes de referencia inicial, si está presente; y
construir una lista de imágenes de referencia del segmento actual de manera que el identificador de la imagen de referencia entre vistas se coloque en la posición determinada de la lista de imágenes de referencia inicial, cuando se determina que el valor para el identificador de capa no es igual a cero.

2. Un procedimiento para codificar los datos de vídeo multivista de acuerdo con un estándar de codificación de vídeo de alta eficiencia, HEVC, comprendiendo el procedimiento:

codificar una pluralidad de elementos de sintaxis en un encabezado de segmento de un segmento actual en una vista actual de los datos de vídeo multivista de acuerdo con una especificación base de HEVC;
codificar un valor para un identificador de capa en el encabezado de segmento del segmento actual, en el que un valor para el identificador de capa que es igual a cero indica que el segmento actual forma parte de una vista base, y en el que un valor para el identificador de capa que es no igual a cero indica que el segmento actual forma parte de una vista no base (202, 208); y solo cuando el segmento actual forma parte de una vista no base:

codificar uno o más elementos de sintaxis adicionales en el encabezado de segmento del segmento actual de acuerdo con una extensión de codificación de vídeo multivista (MV-HEVC) a la especificación de base de HEVC (210), en el que uno o más elementos de sintaxis adicionales incluyen un elemento de sintaxis representativo de si las imágenes de referencia entre vistas se colocan al comienzo de una lista de imágenes de referencia inicial del segmento actual, inmediatamente después de las imágenes de referencia temporal con valores de recuento de orden de imágenes (POC) más pequeños que un valor de POC actual para la lista de imágenes de referencia inicial, inmediatamente después de las imágenes de referencia temporales con valores de recuento de orden de imágenes (POC) mayores que un valor de POC actual para la lista de imágenes de referencia inicial, o inmediatamente después de las imágenes de referencia a largo plazo en la lista de imágenes de referencia inicial, si está presente; y
construir una lista de imágenes de referencia del segmento actual de manera que el identificador de la imagen de referencia entre vistas se coloque en la posición determinada de la lista de imágenes de referencia inicial, cuando el segmento actual forma parte de una vista no base.

3. Un dispositivo (14, 30) para decodificar datos de vídeo multivista de acuerdo con un estándar de codificación de vídeo de alta eficiencia, HEVC, comprendiendo el dispositivo:

los medios para decodificar una pluralidad de elementos de sintaxis en un encabezado de segmento de un segmento actual en una vista actual de los datos de vídeo multivista de acuerdo con una especificación base de HEVC;
los medios para decodificar un valor para un identificador de capa en el encabezado de segmento del segmento actual;
los medios para determinar si el valor para el identificador de capa es igual a cero, en el que un valor para el identificador de capa que es igual a cero indica que el segmento actual forma parte de una vista base, y en el que un valor para el identificador de capa que no es igual a cero indica que el segmento actual forma parte de una vista no base;
los medios para decodificar uno o más elementos de sintaxis adicionales en el encabezado de segmento del segmento actual de acuerdo con una extensión de codificación de vídeo multivista (MV-HEVC) a la

especificación de base HEVC, solo cuando se determina que el valor del identificador de capa no es igual a cero, en el que uno o más elementos de sintaxis adicionales incluyen un elemento de sintaxis representativo de si las imágenes de referencia entre vistas se colocan al comienzo de una lista de imágenes de referencia inicial del segmento actual, inmediatamente después de las imágenes de referencia temporal con valores de recuento de orden de imágenes (POC) más pequeños que un valor de POC actual para la lista de imágenes de referencia inicial, inmediatamente después de las imágenes de referencia temporales con valores de recuento de orden de imágenes (POC) más grandes que un valor de POC actual para la lista de imágenes de referencia inicial, o inmediatamente después de las imágenes de referencia a largo plazo en la lista de imágenes de referencia inicial, si está presente; y los medios para construir una lista de imágenes de referencia del segmento actual de manera que el identificador de la imagen de referencia entre vistas se ubique en la posición determinada de la lista de imágenes de referencia inicial, cuando se determina que el valor del identificador de capa no es igual a cero.

4. Un dispositivo (12, 20) para codificar datos de vídeo multivista de acuerdo con un estándar de codificación de vídeo de alta eficiencia, HEVC, comprendiendo el dispositivo:

los medios para codificar una pluralidad de elementos de sintaxis en un encabezado de segmento de un segmento actual en una vista actual de los datos de vídeo multivista de acuerdo con una especificación base de HEVC;
los medios para codificar un valor para un identificador de capa en el encabezado de segmento del segmento actual, en el que un valor para el identificador de capa que es igual a cero indica que el segmento actual forma parte de una vista base y un valor para el identificador de capa que es no igual a cero indica que el segmento actual forma parte de una vista no base;
los medios para codificar uno o más elementos de sintaxis adicionales en el encabezado de segmento del segmento actual de acuerdo con una extensión de codificación de vídeo multivista (MV-HEVC) a la especificación de base HEVC, cuando el segmento actual forma parte de una vista no base, en el que uno o más elementos de sintaxis adicionales incluyen un elemento de sintaxis representativo de si las imágenes de referencia entre vistas se colocan al comienzo de una lista de imágenes de referencia inicial del segmento actual, inmediatamente después de las imágenes de referencia temporal con valores de recuento de orden de imágenes (POC) más pequeños que un valor de POC actual para la lista de imágenes de referencia inicial, inmediatamente después de las imágenes de referencia temporales con valores de recuento de orden de imágenes (POC) más grandes que un valor de POC actual para la lista de imágenes de referencia inicial, o inmediatamente después de las imágenes de referencia a largo plazo en la lista de imágenes de referencia inicial, si está presente; y
los medios para construir una lista de imágenes de referencia del segmento actual de manera que el identificador de la imagen de referencia entre vistas se coloque en la posición determinada de la lista de imágenes de referencia inicial, cuando el segmento actual forma parte de una vista no base.

5. Un medio de almacenamiento legible por ordenador que tiene almacenadas en el mismo instrucciones que, cuando se ejecutan, hacen que un procesador realice el procedimiento de acuerdo con cualquiera de las reivindicaciones de la 1-2.

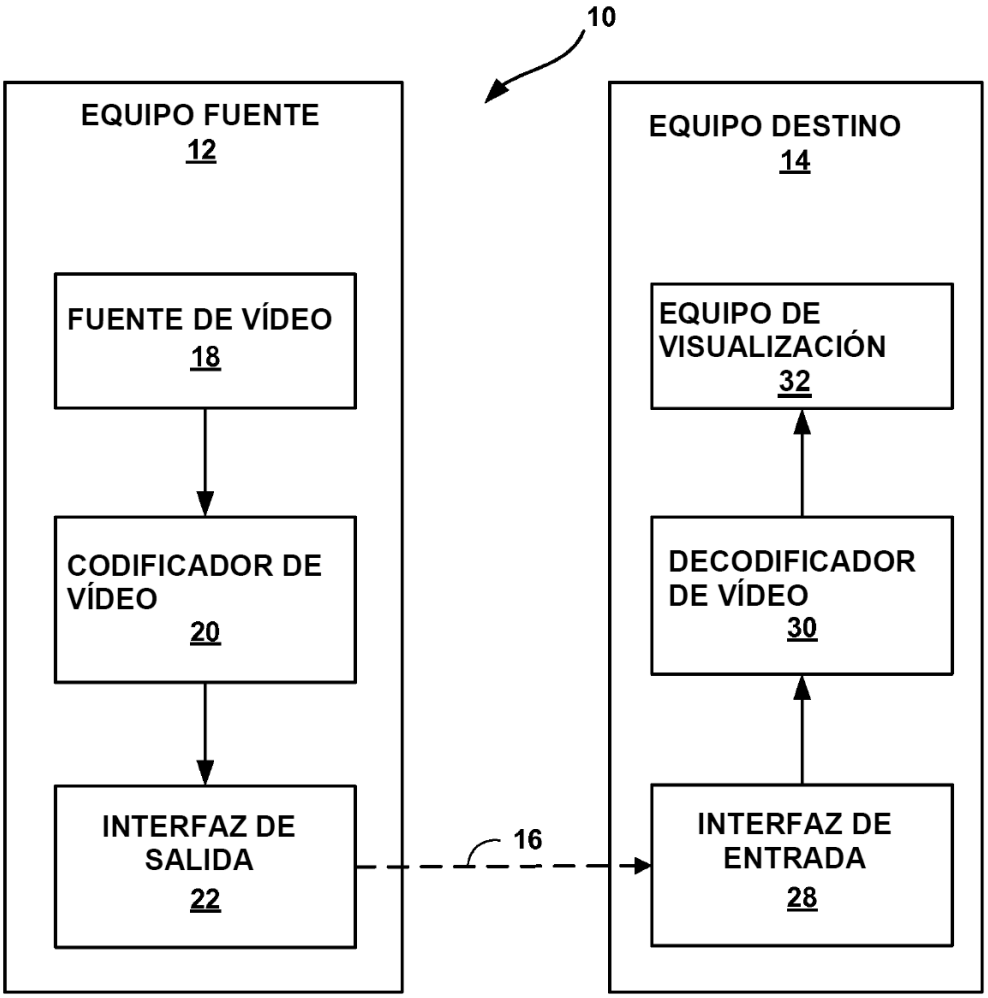
10
EQUIPO FUENTE
12
FUENTE DE VÍDEO
18
CODIFICADOR DE VÍDEO
20
INTERFAZ DE SALIDA
22
16
EQUIPO DESTINO
14
EQUIPO DE VISUALIZACIÓN
32
DECODIFICADOR DE VÍDEO
30
INTERFAZ DE ENTRADA
28

Figura 1

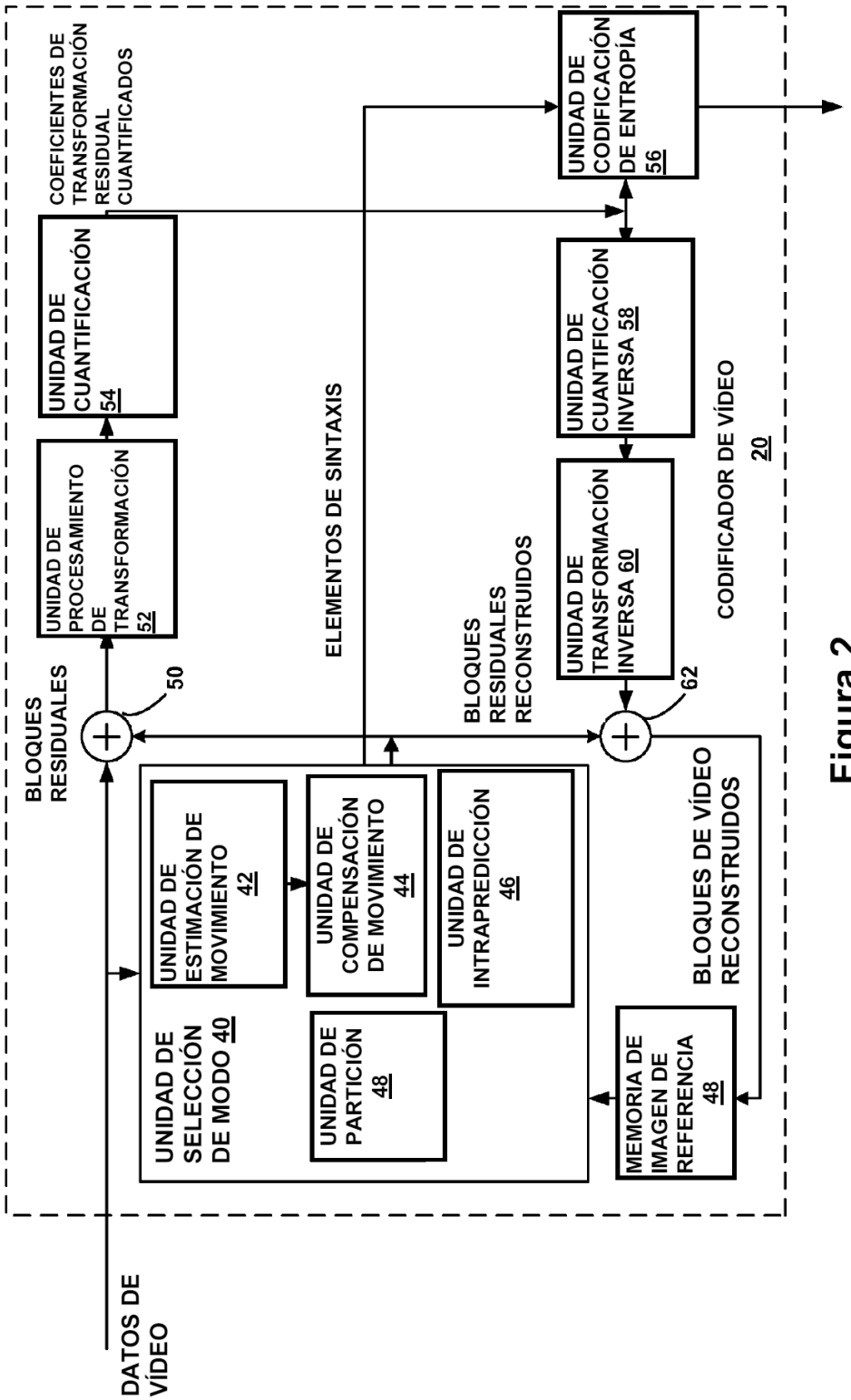
DATOS DE VÍDEO
BLOQUES RESIDUALES
50
UNIDAD DE PROCESAMIENTO DE TRANSFORMACIÓN 52
UNIDAD DE CUANTIFICACIÓN 54
COEFICIENTES DE TRANSFORMACIÓN RESIDUAL CUANTIFICADOS
UNIDAD DE CODIFICACIÓN DE ENTROPÍA 56
ELEMENTOS DE SINTAXIS
UNIDAD DE CUANTIFICACIÓN INVERSA 58
UNIDAD DE TRANSFORMACIÓN INVERSA 60
62
BLOQUES RESIDUALES RECONSTRUIDOS
CODIFICADOR DE VÍDEO 20
UNIDAD DE SELECCIÓN DE MODO 40
UNIDAD DE ESTIMACIÓN DE MOVIMIENTO 42
UNIDAD DE COMPENSACIÓN DE MOVIMIENTO 44
UNIDAD DE INTRAPREDICCIÓN 46
UNIDAD DE PARTICIÓN 48
MEMORIA DE IMAGEN DE REFERENCIA 48
BLOQUES DE VÍDEO RECONSTRUIDOS

Figura 2

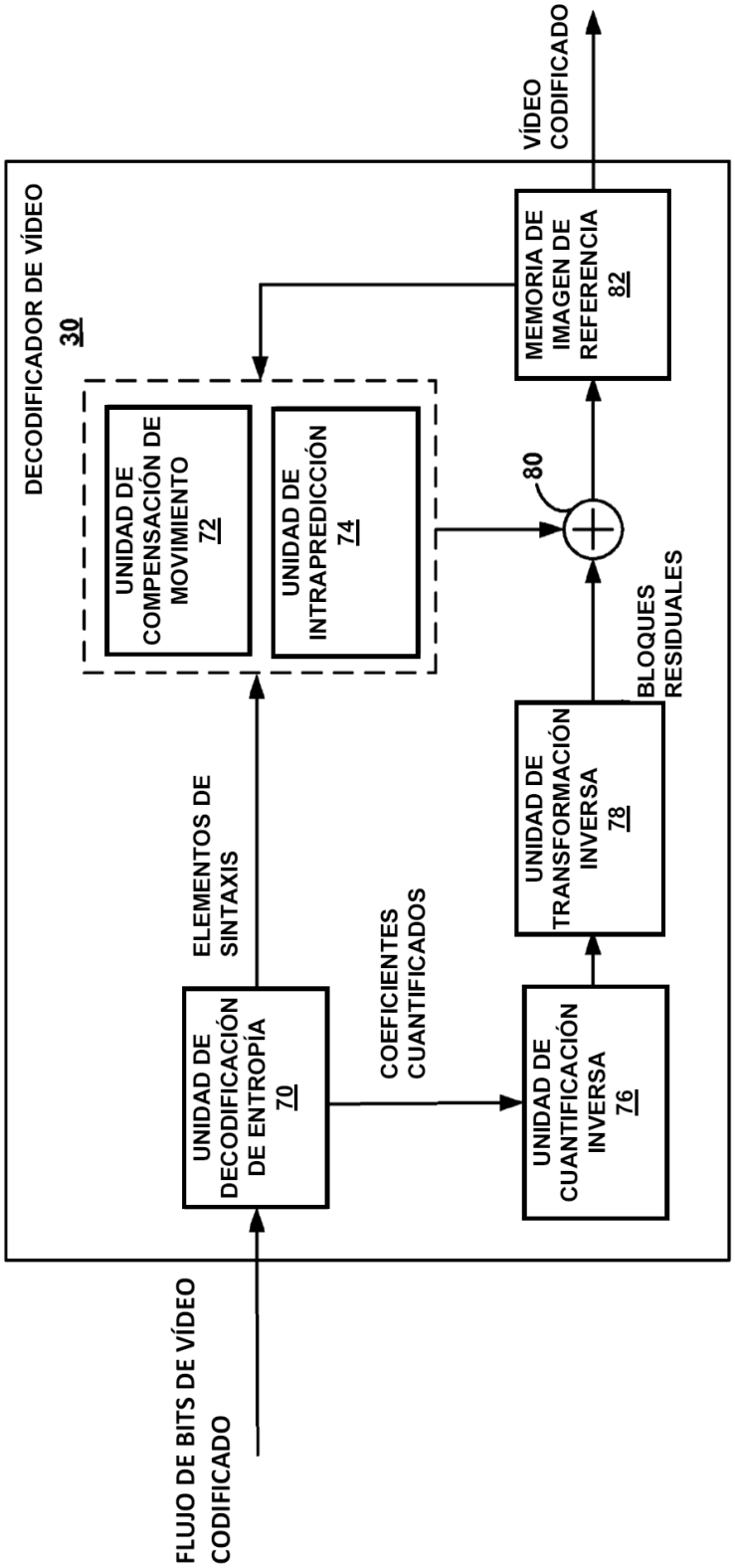
FLUJO DE BITS DE VÍDEO CODIFICADO
DECODIFICADOR DE VÍDEO
30
UNIDAD DE DECODIFICACIÓN DE ENTROPÍA
70
ELEMENTOS DE SINTAXIS
COEFICIENTES CUANTIFICADOS
UNIDAD DE CUANTIFICACIÓN INVERSA
76
UNIDAD DE TRANSFORMACIÓN INVERSA
78
BLOQUES RESIDUALES
80
UNIDAD DE COMPENSACIÓN DE MOVIMIENTO
72
UNIDAD DE INTRAPREDICCIÓN
74
MEMORIA DE IMAGEN DE REFERENCIA
82
VÍDEO CODIFICADO

Figura 3

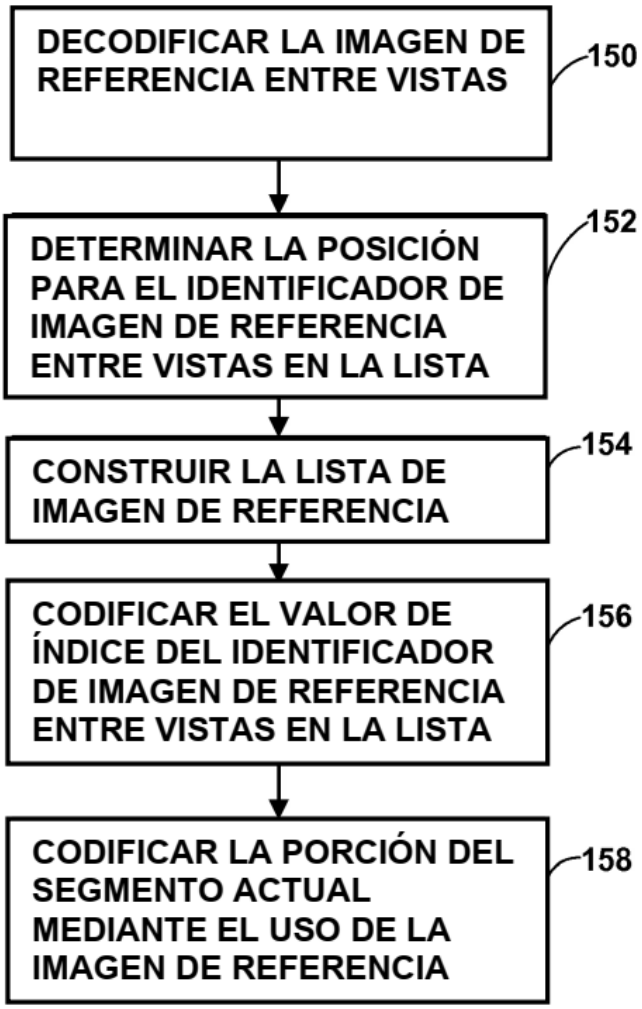
DECODIFICAR LA IMAGEN DE REFERENCIA ENTRE VISTAS
150
DETERMINAR LA POSICIÓN PARA EL IDENTIFICADOR DE IMAGEN DE REFERENCIA ENTRE VISTAS EN LA LISTA
152
CONSTRUIR LA LISTA DE IMAGEN DE REFERENCIA
154
CODIFICAR EL VALOR DE ÍNDICE DEL IDENTIFICADOR DE IMAGEN DE REFERENCIA ENTRE VISTAS EN LA LISTA
156
CODIFICAR LA PORCIÓN DEL SEGMENTO ACTUAL MEDIANTE EL USO DE LA IMAGEN DE REFERENCIA
158

Figura 4

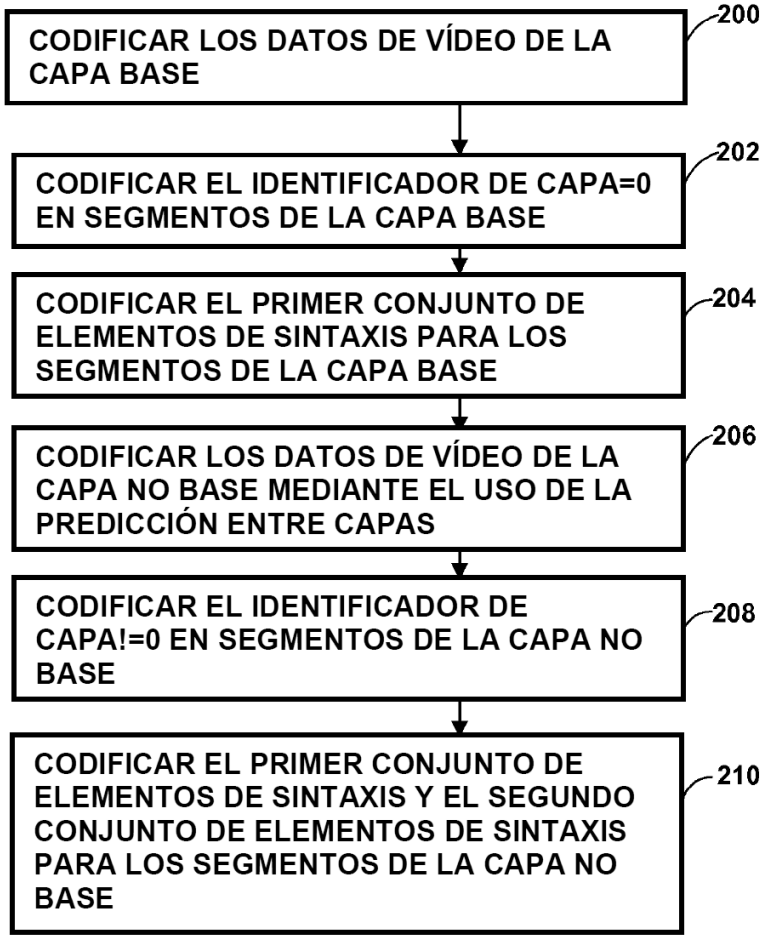
CODIFICAR LOS DATOS DE VÍDEO DE LA CAPA BASE
200
CODIFICAR EL IDENTIFICADOR DE CAPA=0 EN SEGMENTOS DE LA CAPA BASE
202
CODIFICAR EL PRIMER CONJUNTO DE ELEMENTOS DE SINTAXIS PARA LOS SEGMENTOS DE LA CAPA BASE
204
CODIFICAR LOS DATOS DE VÍDEO DE LA CAPA NO BASE MEDIANTE EL USO DE LA PREDICCIÓN ENTRE CAPAS
206
CODIFICAR EL IDENTIFICADOR DE CAPA!=0 EN SEGMENTOS DE LA CAPA NO BASE
208
CODIFICAR EL PRIMER CONJUNTO DE ELEMENTOS DE SINTAXIS Y EL SEGUNDO CONJUNTO DE ELEMENTOS DE SINTAXIS PARA LOS SEGMENTOS DE LA CAPA NO BASE
210

Figura 5

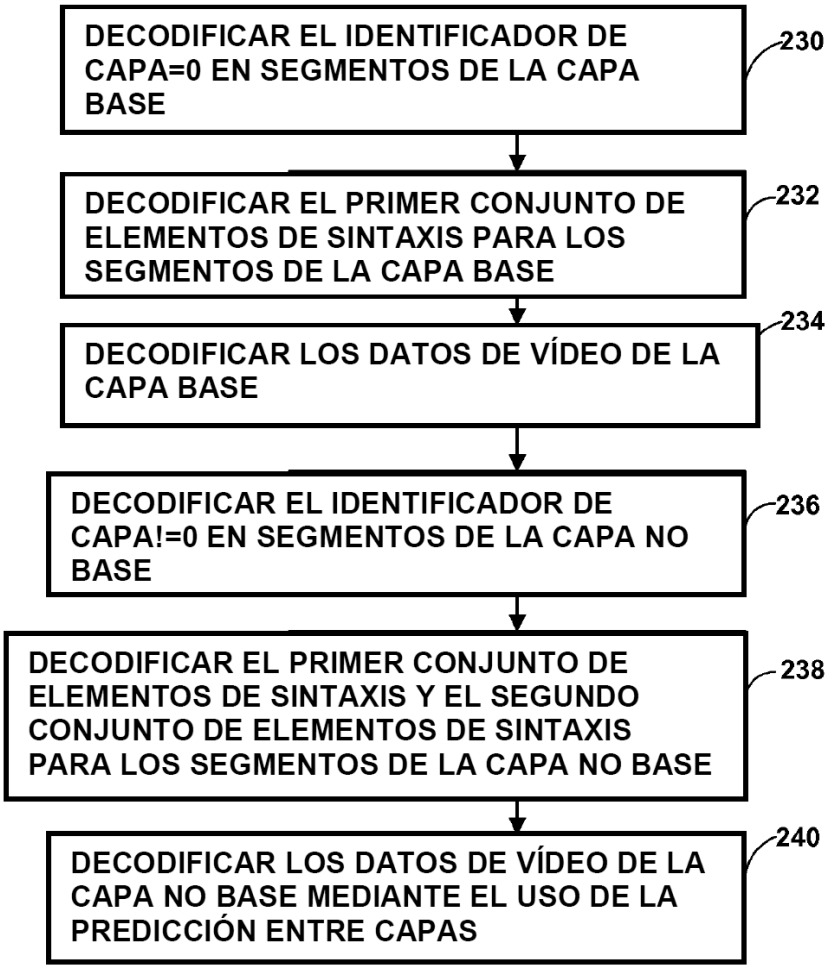
DECODIFICAR EL IDENTIFICADOR DE CAPA=0 EN SEGMENTOS DE LA CAPA BASE
230
DECODIFICAR EL PRIMER CONJUNTO DE ELEMENTOS DE SINTAXIS PARA LOS SEGMENTOS DE LA CAPA BASE
232
DECODIFICAR LOS DATOS DE VÍDEO DE LA CAPA BASE
234
DECODIFICAR EL IDENTIFICADOR DE CAPA!=0 EN SEGMENTOS DE LA CAPA NO BASE
236
DECODIFICAR EL PRIMER CONJUNTO DE ELEMENTOS DE SINTAXIS Y EL SEGUNDO CONJUNTO DE ELEMENTOS DE SINTAXIS PARA LOS SEGMENTOS DE LA CAPA NO BASE
238
DECODIFICAR LOS DATOS DE VÍDEO DE LA CAPA NO BASE MEDIANTE EL USO DE LA PREDICCIÓN ENTRE CAPAS
240

Figura 6