

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 657 494**

51 Int. Cl.:

**H04N 19/70** (2014.01)

**H04N 19/58** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **19.06.2013 PCT/US2013/046529**

87 Fecha y número de publicación internacional: **03.01.2014 WO14004201**

96 Fecha de presentación y número de la solicitud europea: **19.06.2013 E 13737467 (4)**

97 Fecha y número de publicación de la concesión europea: **08.11.2017 EP 2868083**

54 Título: **Acceso aleatorio y señalización de imágenes de referencia a largo plazo en la codificación de vídeo**

30 Prioridad:

**28.06.2012 US 201261665862 P**

**29.06.2012 US 201261666688 P**

**04.03.2013 US 201313784006**

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

**05.03.2018**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)**

**5775 Morehouse Drive**

**San Diego, CA 92121, US**

72 Inventor/es:

**RAMASUBRAMONIAN, ADARSH KRISHNAN;**

**WANG, YE-KUI;**

**JOSHI, RAJAN LAXMAN y**

**CHEN, YING**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 657 494 T3

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

## DESCRIPCIÓN

Acceso aleatorio y señalización de imágenes de referencia a largo plazo en la codificación de vídeo

5 **CAMPO TÉCNICO**

**[0001]** Esta divulgación se refiere, en general, a la codificación de vídeo.

10 **ANTECEDENTES**

10 **[0002]** Las capacidades del vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluyendo  
 15 televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales  
 personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos,  
 cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos,  
 20 consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes",  
 dispositivos de videoconferencia, dispositivos de transmisión de flujo de vídeo y similares. Los dispositivos de vídeo  
 digital implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2,  
 MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de  
 25 Codificación de Vídeo de Alta Eficacia (HEVC), actualmente en desarrollo y las ampliaciones de tales normas. Un  
 borrador reciente de la inminente norma HEVC, denominado "Borrador 7 de trabajo de la HEVC" o "WD7", se  
 describe en el documento HCTVC-I1003, de Bross y col., titulado "High Efficiency vídeo Coding (HEVC) Text  
 Specification Draft 7" ["Especificación textual de la Codificación de Vídeo de Alta Eficacia (HEVC), Borrador 7"],  
 Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC  
 JTC1/SC29/WG11, novena conferencia: Ginebra, Suiza, 27 de abril de 2012 a 7 de mayo de 2012, que, a partir del 5  
 de febrero de 2013, puede descargarse desde [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/documents/9\\_Geneva/wg11/JCTVC-I1003-v5.zip](http://phenix.it-sudparis.eu/jct/doc_end_user/documents/9_Geneva/wg11/JCTVC-I1003-v5.zip). Los dispositivos de vídeo pueden  
 transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficazmente, implementando  
 tales técnicas de codificación de vídeo.

30 **[0003]** Las técnicas de codificación de vídeo incluyen la predicción espacial (intra-imagen) y/o la predicción temporal  
 (entre imágenes) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación  
 de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de  
 vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques arbolados, unidades  
 35 arboladas de codificación, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un  
 fragmento intra-codificado (I) de una imagen se codifican mediante la predicción espacial con respecto a las  
 muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un fragmento inter-  
 codificado (P o B) de una imagen pueden usar la predicción espacial con respecto a las muestras de referencia en  
 bloques vecinos en la misma imagen, o la predicción temporal con respecto a las muestras de referencia en otras  
 40 imágenes de referencia. Las imágenes pueden denominarse tramas y las imágenes de referencia pueden  
 denominarse tramas de referencia.

**[0004]** La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a  
 45 codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque  
 predictivo. Un bloque inter-codificado se codifica de acuerdo a un vector de movimiento que apunta a un bloque de  
 muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el  
 bloque codificado y el bloque predictivo. Un bloque intra-codificado se codifica de acuerdo a una modalidad de intra-  
 50 codificación y a los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde  
 el dominio de los píxeles al dominio de las transformaciones, dando como resultado unos coeficientes de  
 transformación residuales, que posteriormente se pueden cuantizar. Los coeficientes de transformación cuantizados,  
 dispuestos inicialmente en una formación bidimensional, pueden escanearse con el fin de generar un vector  
 unidimensional de coeficientes de transformación, y puede aplicarse la codificación por entropía para lograr aún más  
 compresión. El documento de SULLIVAN (MICROSOFT) G J: "Imágenes de CRA con enlaces rotos", 100.  
 REUNIÓN MPEG; 30-4-2012 a 4-5-2012; GINEBRA; (GRUPO DE EXPERTOS EN IMÁGENES EN MOVIMIENTO O  
 55 ISO / IEC JTC1 / SC29 / WG11) "nº m24653, 7 de mayo de 2012 (2012-05-07), divulga la codificación de las  
 imágenes principales de un punto de acceso aleatorio de CRA.

**RESUMEN**

60 **[0005]** En general, esta divulgación describe técnicas para dar soporte al acceso aleatorio y la señalización de  
 imágenes de referencia a largo plazo en la codificación de vídeo. Esta divulgación propone varias restricciones para  
 dar soporte al acceso aleatorio basado en varios tipos de imágenes. En un ejemplo, todas las imágenes etiquetadas  
 para descartar (TFD) para una imagen de punto de acceso aleatorio (RAP) pueden tener un valor de orden de  
 65 visualización que es anterior al valor de orden de visualización para todas las imágenes principales decodificables  
 (DLP) para el RAP. Esta divulgación también propone una restricción en la que se evita el entrelazado de imágenes  
 principales con "imágenes de cola" en orden de decodificación. Es decir, de acuerdo a las técnicas de esta  
 divulgación, los codificadores de vídeo pueden asegurar que todas las imágenes principales (incluyendo tanto

imágenes TFD y DLP) tengan valores de orden de decodificación anteriores a los valores de orden de decodificación de imágenes que tengan tanto valores de orden de visualización mayores que el valor de orden de visualización de la imagen de RAP correspondiente, como valores de orden de decodificación mayores que los valores de orden de decodificación de la imagen de RAP correspondiente. El orden de visualización también puede denominarse orden de salida. Las imágenes TFD también pueden denominarse imágenes principales salteadas de acceso aleatorio (RASL), y las imágenes DLP también pueden denominarse imágenes principales decodificables de acceso aleatorio (RADL).

**[0006]** En un ejemplo, un procedimiento de codificación de datos de vídeo incluye codificar una imagen de punto de acceso aleatorio (RAP); y codificar una o más imágenes principales decodificables (DLP) para la imagen de RAP de manera que todas las imágenes que estén destinadas a descartarse precedan a las DLP asociadas a la imagen de RAP en orden de visualización.

**[0007]** En otro ejemplo, un dispositivo para codificar datos de vídeo incluye un codificador de vídeo configurado para codificar una imagen de punto de acceso aleatorio (RAP); y codificar una o más imágenes principales decodificables (DLP) para la imagen de RAP de manera que todas las imágenes que estén destinadas a descarte precedan a las DLP asociadas a la imagen de RAP en orden de visualización.

**[0008]** En otro ejemplo, un aparato para la codificación de datos de vídeo incluye medios para la codificación de una imagen de punto de acceso aleatorio (RAP); y medios para codificar una o más imágenes principales decodificables (DLP) para la imagen de RAP de manera que todas las imágenes que estén destinadas a descarte precedan a las DLP asociadas a la imagen de RAP en orden de visualización.

**[0009]** En otro ejemplo, un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando son ejecutadas por uno o más procesadores, provocan que los uno o más procesadores codifiquen una imagen de punto de acceso aleatorio (RAP); y codifiquen una o más imágenes principales decodificables (DLP) para la imagen de RAP de manera que todas las imágenes que estén destinadas a descarte precedan a las DLP asociadas a la imagen de RAP en orden de visualización.

**[0010]** En otro ejemplo, un procedimiento de codificación de datos de vídeo incluye codificar una imagen de punto de acceso aleatorio (RAP) y codificar una o más imágenes principales decodificables (DLP) para la imagen de RAP de modo que todas las imágenes que estén destinadas a descarte precedan a las DLP asociadas a la imagen de RAP en orden de visualización.

**[0011]** En otro ejemplo, un dispositivo para codificar datos de vídeo incluye un codificador de vídeo configurado para codificar una imagen de punto de acceso aleatorio (RAP); y codificar una o más imágenes principales decodificables (DLP) para la imagen de RAP de manera que todas las imágenes que estén destinadas a descarte precedan a las DLP asociadas a la imagen de RAP en orden de visualización.

**[0012]** En otro ejemplo, un aparato para codificar datos de vídeo incluye medios para codificar una imagen de punto de acceso aleatorio (RAP) y medios para la codificación de una o más imágenes principales decodificables (DLP) para la imagen de RAP de modo que todas las imágenes que estén destinadas a descarte precedan a las DLP asociadas a la imagen de RAP en orden de visualización.

**[0013]** En otro ejemplo, un medio de almacenamiento legible por ordenador almacena instrucciones que, cuando son ejecutadas por uno o más procesadores, provocan que los uno o más procesadores codifiquen una imagen de punto de acceso aleatorio (RAP); y codifiquen una o más imágenes principales decodificables (DLP) para la imagen de RAP de manera que todas las imágenes que estén destinadas a descarte precedan a las DLP asociadas a la imagen de RAP en orden de visualización.

**[0014]** Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y los dibujos, y a partir de las reivindicaciones.

**BREVE DESCRIPCIÓN DE LOS DIBUJOS**

**[0015]**

La FIG. 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo que puede utilizar técnicas para señalar imágenes de referencia a largo plazo en cabeceras de fragmentos.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas para señalar imágenes de referencia a largo plazo en cabeceras de fragmentos.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar técnicas para señalar imágenes de referencia a largo plazo en cabeceras de fragmentos.

La FIG. 4 es un diagrama conceptual que ilustra una secuencia de imágenes de vídeo codificadas.

5 La FIG. 5 es un diagrama de bloques que ilustra un conjunto ejemplar de dispositivos que forman parte de una red.

La FIG. 6 es un diagrama de flujo que ilustra una operación ejemplar para codificar imágenes de punto de acceso aleatorio (RAP) de acuerdo a técnicas de esta divulgación.

10 La FIG. 7 es un diagrama de flujo que ilustra una operación ejemplar para codificar imágenes de RAP de acuerdo a las técnicas de esta divulgación.

La FIG. 8 es un diagrama de flujo que ilustra una operación ejemplar para codificar imágenes de RAP de acuerdo a las técnicas de esta divulgación.

15 La FIG. 9 es un diagrama de flujo que ilustra una operación ejemplar para codificar imágenes de RAP de acuerdo a las técnicas de esta divulgación.

20 La FIG. 10 es un diagrama de flujo que ilustra una operación ejemplar para codificar imágenes de RAP de acuerdo a las técnicas de esta divulgación.

La FIG. 11 es un diagrama de flujo que ilustra una operación ejemplar para codificar imágenes de RAP de acuerdo a las técnicas de esta divulgación.

25 **DESCRIPCIÓN DETALLADA**

[0016] En general, los datos de vídeo se representan por una secuencia de imágenes que son capturadas o exhibidas en rápida sucesión. Se debería entender que, en algunos ejemplos, se pueden generar imágenes o partes de las imágenes, por ejemplo, usando gráficos de ordenador, en lugar de (o además de) ser capturadas. El orden en que se han de exhibir las imágenes (que puede o no ser el mismo que el orden en que se capturaron o generaron las imágenes) puede diferir del orden en que las imágenes codificadas se incluyen en el flujo de bits. El orden de visualización de las imágenes dentro de una secuencia de vídeo codificada puede representarse mediante valores de recuento de orden de imágenes (POC), mientras que el orden de codificación de las imágenes puede representarse mediante valores de número de trama (núm\_trama).

35 [0017] La codificación de imágenes generalmente implica aprovechar los datos redundantes que aparecen en las imágenes. Por ejemplo, la codificación espacial, también denominada intra-codificación-predictiva, aprovecha las redundancias que se producen entre los bloques espacialmente vecinos de valores de píxel de una imagen común. Como otro ejemplo, la codificación temporal, también conocida como inter-codificación-predictiva, aprovecha las redundancias que se producen entre diferentes imágenes que se capturan, generan o exhiben en diferentes instancias temporales. Más en particular, una imagen se puede dividir en fragmentos, que se pueden designar para intra-predicción o inter-predicción. Además, cada fragmento puede dividirse en bloques (por ejemplo, las máximas unidades de codificación (LCU), también denominadas unidades de árbol de codificación), y cada uno de los bloques puede ser adicionalmente dividido o codificado basándose en si el fragmento correspondiente ha sido designado para la intra-predicción o la inter-predicción.

40 [0018] Con respecto al ejemplo de la inter-codificación-predictiva, los bloques de una imagen que se está codificando actualmente (también mencionada como una "imagen actual") pueden ser predichos a partir de una imagen de referencia. El documento WD7 define una "imagen de referencia" como una imagen con un indicador\_ref\_nal igual a 1. El indicador\_ref\_nal es un elemento sintáctico de una unidad de capa de abstracción de red (NAL) que indica si los datos incluidos en la unidad de NAL deben tratarse como una imagen de referencia. El documento WD7 también provee que una imagen de referencia contiene muestras (es decir, valores de píxel) que pueden usarse para la inter-predicción en el proceso de decodificación de imágenes posteriores en orden de decodificación.

55 [0019] Como se ha presentado anteriormente, en general, los datos de vídeo incluyen varios tipos de imágenes. Por ejemplo, los datos de vídeo incluyen puntos de acceso aleatorio (RAP) que se pueden usar para comenzar un flujo de datos de vídeo. Los RAP son imágenes que están codificadas independientemente, por ejemplo, como imágenes codificadas por intra-predicción (imágenes-I). Los RAP incluyen imágenes de actualización instantánea de decodificador (IDR), imágenes de acceso de enlace roto (BLA), imágenes limpias de acceso aleatorio (CRA) e imágenes de actualización gradual de decodificación (GDR). Otros tipos de imágenes incluyen imágenes destinadas a descarte (TFD), imágenes principales decodificables (DLP) e imágenes de cola. En general, las DLP (como todas las imágenes principales) son imágenes que tienen un orden de salida que es anterior a un orden de salida de un RAP correspondiente, pero un orden de decodificación que es posterior al orden de decodificación del RAP correspondiente. Sin embargo, las DLP son imágenes principales que son decodificables sin recuperar los datos que preceden al RAP correspondiente en el orden de decodificación. Las imágenes de cola, por otro lado, tienen un

orden de salida que es posterior al orden de salida del RAP correspondiente, así como un orden de decodificación que es posterior al orden de decodificación del RAP correspondiente. Las imágenes que son TFD no se pueden decodificar debidamente cuando se realiza un acceso aleatorio desde un RAP correspondiente. En general, esto se debe a que se predice una imagen TFD a partir de los datos que preceden al RAP correspondiente en el orden de decodificación. Como el acceso aleatorio desde un RAP omite recuperar datos anteriores al RAP en orden de decodificación, las imágenes TFD, que dependen de datos anteriores al RAP en orden de decodificación, no son decodificables correctamente cuando el RAP se usa como punto de acceso aleatorio y, por lo tanto, puede ser descartado sin ser decodificado. Las imágenes TFD "dependen" de datos anteriores en el sentido de que se predicen en relación con una o más imágenes que aparecen anteriormente, en el orden de decodificación, al RAP.

**[0020]** A las unidades de NAL se pueden asignar valores particulares de tipo de unidad de NAL, por ejemplo, en una cabecera de unidad de NAL, para indicar el tipo de datos incluidos en la unidad de NAL correspondiente. Las unidades de NAL generalmente pueden incluir datos de la capa de codificación de vídeo (VCL), también mencionados como unidades de NAL de VCL, o datos no de la VCL, también mencionados como unidades de NAL no de VCL. Los datos no de VCL incluyen, por ejemplo, conjuntos de parámetros tales como conjuntos de parámetros de imagen (PPS), conjuntos de parámetros de secuencia (SPS) y conjuntos de parámetros de adaptación (APS), así como mensajes de información de mejora suplementaria (SEI). Los datos de VCL generalmente incluyen datos de vídeo codificados. El tipo de unidad de NAL puede proporcionar una indicación de un tipo de datos incluidos en la unidad de NAL, que incluye si los datos codificados de vídeo en la unidad de NAL corresponden a un RAP, y, si es así, si los datos codificados de vídeo corresponden a una imagen de CRA, BLA o IDR. El tipo de unidad de NAL puede indicar si la unidad de NAL también incluye otros tipos de datos. La Tabla 1 a continuación proporciona un ejemplo de algunos valores y datos del tipo de unidad de NAL incluidos en una unidad de NAL correspondiente, como se proveyó en el documento WD7:

**TABLA 1**

nal_unit_type	Contenido de la unidad de NAL y estructura sintáctica RBSP	Clase de tipo de unidad de NAL
1	Fragmento codificado de una imagen no RAP, no TFD ni TLA slice_layer_rbsp ()	VCL
2	Fragmento codificado de una imagen TFD slice_layer_rbsp ()	VCL
3	Fragmento codificado de una imagen TLA no TFD slice_layer_rbsp ()	VCL
4, 5	Fragmento codificado de una imagen CRA slice_layer_rbsp ()	VCL
6, 7	Fragmento codificado de una imagen BLA slice_layer_rbsp ()	VCL
8	Fragmento codificado de una imagen IDR slice_layer_rbsp ()	VCL

**[0021]** Por otra parte, el documento WD7 distingue imágenes de referencia a largo plazo de imágenes de referencia a corto plazo. Por ejemplo, el documento WD7 define una imagen de referencia a largo plazo como una imagen de referencia que está marcada como "utilizada para referencia a largo plazo". El documento WD7 provee que un indicador en un conjunto de parámetros de secuencia (SPS) indica si las imágenes de referencia a largo plazo se señalizan en absoluto para una secuencia de vídeo codificada. De acuerdo al documento WD7, los codificadores de vídeo señalizan los bits menos significativos (LSB) de los valores de POC de las imágenes de referencia a largo plazo en las cabeceras de fragmento de los fragmentos. La señalización de los LSB puede dar como resultado ahorros de bits, en relación con la señalización del valor de POC completo. Los codificadores de vídeo, de acuerdo al documento WD7, pueden señalar los bits más significativos (MSB) de los valores de POC de una imagen de referencia a largo plazo si hay más de una imagen de referencia en una memoria temporal decodificada (DPB) que tiene los mismos bits LSB que la imagen de referencia a largo plazo. Además, de acuerdo al documento WD7, los codificadores de vídeo pueden usar un indicador para indicar si las imágenes de referencia a largo plazo señalizadas pueden ser utilizadas por una imagen actual como referencia.

**[0022]** Una hipótesis para el diseño del conjunto de imágenes de referencia (RPS) de la HEVC es que la obtención del RPS debe ser robusta ante pérdidas de imágenes, mientras los valores de POC relativos puedan establecerse correctamente, tanto para imágenes de referencia a corto plazo (STRP) como para imágenes de referencia a largo plazo (LTRP). Para lograr esto, la obtención del RPS para cualquier imagen puede ser auto-contenida, es decir, no dependiente del estado de la memoria temporal de imágenes decodificadas (DPB). Con este principio de diseño, las STRP, de acuerdo al documento WD7, se señalizan utilizando sus valores de delta de POC, con respecto a la

imagen actual. Por lo tanto, el decodificador puede ser capaz de obtener los valores de POC de estas imágenes de referencia incluso cuando se perdieran algunas imágenes y, por lo tanto, comprobar si está presente cada una de las STRP.

5 **[0023]** Esta divulgación reconoce ciertos problemas en la señalización actual de cabeceras de fragmento de las LTRP en presencia de pérdidas de imágenes del documento WD7. La obtención errónea del RPS es posible en presencia de pérdidas. Esto se debe a señalar solamente la cantidad de los LSB del POC para las LTRP que es igual que para la señalización del valor de POC para cada imagen actual (es decir, `pic_order_cnt_lsb`), aunque en algunos casos los valores de POC completos, usando el elemento sintáctico `delta_poc_msb_cycle_lf`, son  
10 señalizados para las LTRP. Si hay una imagen en la DPB que tiene el mismo LSB que el señalado para la LTRP, entonces podría haber ambigüedad en la obtención del RPS, y podría ocurrir un resultado incorrecto del RPS, como se muestra en los ejemplos que se presentan a continuación. En los ejemplos que se exponen a continuación, se supone que "MaxPicOrderCntLsb" es igual a 256 o, en otras palabras, `log2_max_pic_order_cnt_lsb_minus4` es igual a 4.

15 **[0024]** Como un primer ejemplo para ilustrar los problemas expuestos anteriormente, considérense tres imágenes  $n - 1$ ,  $n$  y  $n + 1$ , que son sucesivas en orden de decodificación. Supongamos que la imagen  $n - 1$  tiene imágenes de referencia con valores de POC de 0 y 256 como las LTRP, y las imágenes  $n$  y  $n + 1$  solo tienen una imagen de referencia con POC 256 como la LTRP. Los valores de POC de las imágenes  $n - 1$ ,  $n$  y  $n + 1$  están todos en el rango de 257 a 511, inclusive. Supongamos también que ninguna de las imágenes  $n - 1$ ,  $n$  y  $n + 1$ , o cualquier otra imagen en la DPB, tiene un valor de LSB igual a 0. Según la señalización del documento WD7 de la HEVC, los elementos sintácticos relevantes de un flujo de bits conforme son como se muestran en la Tabla 2.

TABLA 2

POC (en orden de decodificación)	Las LTRP	Los LSB de las LTRP	<code>delta_poc_msb_present_flag</code>	<code>delta_poc_msb_cycle_lf</code>
...	...	...	...	...
$n - 1$	0, 256	0, 0	1, 1	1, 0
$n$	256	0	1	0
$n + 1$	256	0	0	-
...	...	...	...	...

25 **[0025]** Para la imagen  $n$ , incluso aunque solamente la imagen 256 se señala como la LTRP, cuando se obtiene el RPS de la imagen  $n$ , la imagen 0 todavía estaría en la DPB. Por lo tanto, incluso para la imagen  $n$ , el `delta_poc_msb_present_flag` se fija en 1, y se envía el valor `delta_poc_msb_cycle_lf`. Después de la obtención del RPS para la imagen  $n$ , la imagen 0 se marca como "no utilizada para referencia" y puede eliminarse de la DPB. Para la imagen  $n + 1$ , sin embargo, dado que solo hay una LTRP presente en la DPB, el `delta_poc_msb_present_flag` se fija en 0 y `delta_poc_msb_cycle_lf` no está presente.

30 **[0026]** Según la estructura sintáctica del documento WD7, la tabla anterior corresponderá a un flujo de bits conforme. Sin embargo, si la imagen  $n$  se pierde, entonces la señalización de la LTRP para la imagen  $n + 1$  sería insuficiente para resolver qué imagen de la DPB se debe seleccionar como la LTRP. Tanto 0 como 256 tienen un LSB de 0, y el decodificador no podría resolver el problema. El decodificador necesitaría que el `delta_poc_msb_present_flag` fuese igual a 1 para la imagen  $n + 1$  para identificar de manera unívoca la imagen a utilizar como la única LTRP.

35 **[0027]** Un segundo ejemplo de los problemas expuestos anteriormente, relacionados con el primer ejemplo anterior, es el siguiente. La descripción de la Tabla 3 es similar a la descripción de la Tabla 2, con la excepción de que la imagen 0 se señala como LTRP y la imagen 256 se señala como STRP para la imagen  $n - 1$ .

TABLA 3

POC (en orden de decodificación)	Las LTRP	Las STRP	Los LSB de las LTRP	<code>delta_poc_msb_present_flag</code>	<code>delta_poc_msb_cycle_lf</code>
...	...	...	...	...	...
$n - 1$	0	256	0	1	1
$n$	256	-	0	1	0
$n + 1$	256	-	0	0	-
...	...	...	...	...	...

40 **[0028]** Aquí, de nuevo, si se pierde la imagen  $n$ , el decodificador no sería capaz de identificar correctamente que la

LTRP señalizada como la imagen 256, y el decodificador necesitaría que el delta\_poc\_msb\_present\_flag sea igual a 1 para identificar de forma unívoca la imagen a utilizar como la única LTRP.

[0029] Un tercer ejemplo de los problemas expuestos anteriormente se describe con respecto a la Tabla 4.

5

**TABLA 4**

POC (en orden de decodificación)	Las LTRP	STRP	Los LSB de las LTRP	delta_poc_msb_present_flag	delta_poc_msb_cycle_lt
...	...	...	...	...	...
255	0	-	0	0	
256	0	-	0	0	
257	256	-	0	1	0
258	256	-	0	0	-
...	...	...	...	...	...

[0030] Con respecto al ejemplo de la Tabla 4, supongamos que se recibe una imagen con POC 255, seguida por la imagen 258 (en otras palabras, las imágenes 256 y 257 se perdieron). Después de decodificar una cabecera de fragmento para la imagen 258, la imagen 0 (en lugar de la imagen 256) se marcará como una LTRP. En lugar de deducir que la imagen de referencia 256 no está presente para la imagen 258, el decodificador deduciría incorrectamente que la imagen 0 es la LTRP señalizada y continuaría decodificando (a condición de que otros aspectos del proceso de decodificación funcionen para esta pérdida).

10

[0031] A partir de los ejemplos anteriores, una solución engañosamente sencilla para resolver los problemas anteriores puede parecer ser señalar el valor real completo del POC de las LTRP. También parecería que la obtención del RPS sería auto-contenida con señalización de POC completa. Sin embargo, la posibilidad de que las imágenes de CRA puedan comenzar un flujo de bits, o de que las imágenes de CRA se puedan convertir en una imagen de BLA por medio de un empalmador, descarta la señalización del valor de POC completo. Por ejemplo, considérese una imagen de CRA en un flujo de bits con un valor de POC de 256 (suponiendo aún que MaxPicOrderCntLsb = 256), y que algunas imágenes que siguen a la imagen de CRA en el orden de decodificación tengan la imagen de CRA como una LTRP. Si se señala el POC completo de la imagen de CRA, que es 256, y la imagen de CRA se convierte en una imagen de BLA o inicia un flujo de bits, entonces la LTRP sería incorrectamente obtenida, y el flujo de bits se volverá no conforme. Cuando una imagen de CRA se convierte en una imagen de BLA o inicia un flujo de bits, el decodificador obtiene su valor de POC solo en base a su LSB, de acuerdo a las técnicas del documento WD7. En el ejemplo anterior, se determinará que el POC de la imagen de CRA (que ahora es una imagen de BLA) es 0. Cuando un decodificador busca una LTRP con POC 256, la DPB no contendrá ninguna de esas imágenes porque ahora la imagen de CRA ha obtenido un POC de 0. Esto daría como resultado una "imagen sin referencia" en el RPS y, si la imagen actual no es una imagen TFD asociada a la imagen de CRA, el flujo de bits resultante no sería conforme.

15

20

25

30

[0032] Esta divulgación describe técnicas para dar soporte al acceso aleatorio para datos de vídeo. En algunos casos, estas técnicas pueden usarse para superar los problemas anteriores del documento WD7. Más en particular, esta divulgación propone varias restricciones para dar soporte al acceso aleatorio basándose en varios tipos de imágenes. En primer lugar, de acuerdo a una técnica de esta divulgación, todas las imágenes TFD para una imagen de RAP deberían tener un valor de orden de visualización que sea anterior al valor de orden de visualización para todas las DLP para el RAP. Esta restricción puede mejorar la experiencia de visualización de un espectador al evitar encrespamiento en la velocidad de tramas. Es decir, si hay una o más imágenes que no se pueden decodificar correctamente (es decir, imágenes TFD) y si estas imágenes tienen valores de orden de visualización entremezclados con las DLP, la velocidad de las tramas parecería encrespada, lo que disminuiría la experiencia del usuario. De acuerdo a otra técnica, esta divulgación también propone una restricción en la que se evita el entrelazado de imágenes principales con "imágenes de cola" en el orden de decodificación. Es decir, de acuerdo a las técnicas de esta divulgación, los codificadores de vídeo pueden asegurar que todas las imágenes principales (incluyendo tanto imágenes TFD como las DLP) tengan valores de orden de decodificación anteriores a los valores de orden de decodificación de imágenes que tengan valores de orden de visualización mayores que el valor de orden de visualización de la imagen de RAP correspondiente. Como se ha indicado anteriormente, las "imágenes de cola" son imágenes que siguen a la imagen del RAP, tanto en orden de decodificación como en orden de visualización.

35

40

45

[0033] Como se ha presentado anteriormente, en general, los datos de vídeo incluyen varios tipos de imágenes. Por ejemplo, los datos de vídeo incluyen los RAP que pueden usarse para comenzar un flujo de datos de vídeo. Los RAP son imágenes que están codificadas independientemente, por ejemplo, como imágenes codificadas por intrapredicción (imágenes-I). Los RAP incluyen imágenes de IDR, imágenes de BLA e imágenes de CRA.

50

[0034] Otros tipos de imágenes incluyen TFD, DLP e imágenes de cola. En general, las DLP (como todas las

55

imágenes principales) son imágenes que tienen un orden de salida que es anterior a un orden de salida de la imagen de RAP correspondiente, pero un orden de decodificación que es posterior al orden de decodificación del RAP correspondiente. Sin embargo, las DLP son imágenes principales que se pueden decodificar sin recuperar datos que preceden a la imagen de RAP correspondiente en el orden de decodificación. En otras palabras, las DLP son imágenes principales que se predicen en relación con los datos de una o más imágenes que incluyen y / o siguen el RAP correspondiente. Las imágenes de cola, por otro lado, tienen un orden de salida que es posterior al orden de salida del RAP correspondiente, así como un orden de decodificación que es posterior al orden de decodificación de la imagen de RAP correspondiente.

**[0035]** Las imágenes que son imágenes TFD no pueden decodificarse adecuadamente cuando se realiza un acceso aleatorio desde la imagen de RAP correspondiente. Esto se debe generalmente a una imagen TFD que se predice a partir de datos que preceden a la imagen de RAP correspondiente en el orden de decodificación. Debido a que el acceso aleatorio desde una imagen de RAP omite recuperar datos anteriores a la imagen de RAP, las imágenes TFD que dependen de datos anteriores a la imagen de RAP no se podrán decodificar correctamente y, por lo tanto, se pueden descartar sin decodificar.

**[0036]** Esta divulgación propone una serie de restricciones para dar soporte al acceso aleatorio basado en estos diversos tipos de imágenes. En primer lugar, todas las imágenes TFD de una imagen de RAP deben tener un valor de orden de visualización que sea anterior al valor de orden de visualización para todas las DLP de la imagen de RAP. Esta restricción puede mejorar la experiencia de visualización de un espectador, ya que evita el encrespamiento en la velocidad de tramas. Es decir, si hay una o más imágenes que no se pueden decodificar correctamente (tales como imágenes TFD), si estas imágenes tenían valores de orden de visualización entremezclados con las DLP, la velocidad de tramas parecería encrespada, lo que disminuiría la experiencia del usuario.

**[0037]** Esta divulgación también propone una restricción en la que se impide el entrelazado de las imágenes principales con "imágenes de cola" en el orden de decodificación. Es decir, de acuerdo a las técnicas de esta divulgación, los codificadores de vídeo pueden asegurar que todas las imágenes principales (incluyendo tanto imágenes TFD como las DLP) tengan valores de orden de decodificación anteriores a los valores de orden de decodificación de imágenes que tengan valores de orden de visualización mayores que el valor de orden de visualización de la imagen de RAP correspondiente. Como se ha indicado anteriormente, las "imágenes de cola" son imágenes que siguen a la imagen del RAP, tanto en orden de decodificación como en orden de visualización. Implementando las técnicas de esta divulgación, un codificador de vídeo puede configurarse para codificar una imagen de RAP y codificar una o más DLP para la imagen de RAP, de modo que todas las imágenes que se destinan a descarte precedan a las DLP asociadas a la imagen de RAP en el orden de salida. Del mismo modo, un decodificador de vídeo no necesita intentar decodificar imágenes TFD a continuación de una imagen de RAP que se usa como un punto de acceso aleatorio. En cambio, el decodificador de vídeo puede simplemente analizar sintácticamente los datos para tales imágenes TFD, sin intentar decodificar los datos. Esto puede proporcionar un ahorro de recursos para un dispositivo tal como un decodificador de vídeo. Por ejemplo, estas técnicas pueden ahorrar energía de la batería y / o mejorar la eficacia del procesamiento. Al no intercalar las imágenes principales y de cola en el orden de decodificación, un sistema (como un MANE o un decodificador de vídeo) puede identificar fácilmente las imágenes principales asociadas a la imagen de RAP, sin tener que analizar sintácticamente más allá de la primera imagen de cola que sigue a la Imagen de RAP asociada. La identificación de las imágenes principales puede ser útil para decidir la salida en el acceso aleatorio, porque la imagen principal se produce antes del RAP en orden de visualización y puede ser útil para eliminar las imágenes principales si un sistema intermedio está configurado para hacerlo. Además, como se ha indicado anteriormente, estas técnicas pueden conducir a una experiencia de usuario mejorada al evitar el encrespamiento de la velocidad de tramas.

**[0038]** La FIG. 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo 10 que puede utilizar técnicas para la señalización de imágenes de referencia a largo plazo en cabeceras de fragmentos. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados, a decodificar en un momento posterior mediante un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 mediante un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, equipos telefónicos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, un dispositivo de transmisión de flujo de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

**[0039]** El dispositivo de destino 14 puede recibir los datos de vídeo codificados a decodificar mediante el medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación que habilita al dispositivo de origen 12 para transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo a una norma de comunicación, tal como



un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global tal como Internet.

5 El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

**[0040]** En algunos ejemplos, pueden emitirse datos codificados desde la interfaz de salida 22 hasta un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados del dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera entre una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco rígido, discos Blu-ray, discos DVD, discos CD-ROM, memoria flash, memoria volátil o no volátil u otros medios adecuados cualesquiera de almacenamiento digital, para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda almacenar los datos de vídeo codificados, generados por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados del dispositivo de almacenamiento, mediante flujo o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los ejemplos de servidores de archivos incluyen un servidor de la Red (por ejemplo, para una página de la Red), un servidor de FTP, dispositivos de almacenamiento anexos a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados mediante cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión por flujo, una transmisión de descarga o una combinación de ambas.

**[0041]** Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de cualquiera entre una diversidad de aplicaciones de multimedios, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por flujo de Internet, tales como el flujo adaptativo dinámico sobre HTTP (DASH), el vídeo digital que se codifica en un medio de almacenamiento de datos, la decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para prestar soporte a la transmisión de vídeo unidireccional o bidireccional a fin de prestar soporte a aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

**[0042]** En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. Según esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede estar configurado para aplicar las técnicas para la señalización de imágenes de referencia a largo plazo en cabeceras de fragmentos. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde un origen de vídeo externo 18, tal como una cámara externa. Asimismo, el dispositivo de destino 14 puede mantener interfaces con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

**[0043]** El sistema 10 ilustrado de la FIG. 1 es simplemente un ejemplo. Las técnicas para señalar imágenes de referencia a largo plazo en cabeceras de fragmentos pueden ser realizadas por cualquier dispositivo de codificación y / o decodificación de vídeo digital. Aunque, en general, las técnicas de esta divulgación se llevan a cabo mediante un dispositivo de codificación de vídeo, las técnicas también pueden llevarse a cabo mediante un codificador / decodificador de vídeo, denominado habitualmente "CÓDEC". Además, las técnicas de esta divulgación también pueden llevarse a cabo mediante un pre-procesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de tales dispositivos de codificación, en los cuales el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera esencialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y de decodificación de vídeo. Por lo tanto, el sistema 10 puede dar soporte a una transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para la transmisión por flujo de vídeo, la reproducción de vídeo, la difusión de vídeo o la videotelefonía.

**[0044]** El origen de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo grabado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenidos de vídeo. Como una alternativa adicional, el origen de vídeo 18 puede generar datos, basados en gráficos de ordenador, como el vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con

cámara o videoteléfonos. Sin embargo, como se ha mencionado anteriormente, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas. En cada caso, el vídeo capturado, pre-capturado o generado por ordenador puede ser codificado por el codificador de vídeo 20. La información de vídeo codificada puede ser entonces emitida por la interfaz de salida 22 hacia un medio legible por ordenador 16.

**[0045]** El medio legible por ordenador 16 puede incluir medios transitorios, tales como una difusión inalámbrica o una transmisión de red cableada, o medios de almacenamiento (es decir, medios de almacenamiento no transitorio), tales como un disco duro, una unidad de memoria flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, mediante transmisión por red. De manera similar, un dispositivo informático de una instalación de producción de un medio, tal como una instalación de grabación de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y producir un disco que contiene los datos de vídeo codificados. Por lo tanto, puede entenderse que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de varias formas, en varios ejemplos.

**[0046]** La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde el medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también es usada por el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o el procesamiento de bloques y otras unidades codificadas, por ejemplo, grupos de imágenes (GOP). El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario y puede comprender cualquiera entre una variedad de dispositivos de visualización, tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

**[0047]** El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en fase de desarrollo, y pueden ajustarse al modelo de prueba de la HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a otras normas privadas o industriales, tales como la norma ITU-T H.264, alternativamente denominada MPEG-4, Parte 10, codificación de vídeo avanzado (AVC), o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de codificación de vídeo incluyen las de MPEG-2 e ITU-T H.263. Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar integrados, cada uno de ellos, con un codificador y un decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos diferentes. Si procede, las unidades de MUX-DEMUX pueden ajustarse al protocolo de multiplexado ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

**[0048]** La norma ITU-T H.264 / MPEG-4 (AVC) fue formulada por el Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T, junto al Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO / IEC, como el producto de una asociación colectiva conocida como el Equipo de Vídeo Conjunto (JVT). En algunos aspectos, las técnicas descritas en esta divulgación pueden ser aplicadas a dispositivos que se ajustan en general a la norma H.264. La norma H.264 está descrita en la Recomendación ITU-T H.264, Codificación de Vídeo Avanzada, para los servicios audiovisuales genéricos, por el Grupo de Estudio de la ITU-T, y con fecha de marzo de 2005, y que se puede denominar en el presente documento norma H.264 o memoria descriptiva H.264, o la norma o memoria descriptiva H.264/AVC. El Equipo de Vídeo Conjunto (JVT) continúa trabajando en ampliaciones para la norma H.264/MPEG-4 AVC.

**[0049]** El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse, cada uno, como cualquiera entre una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de estos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio, legible por ordenador, y ejecutar las instrucciones en hardware mediante uno o más procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

**[0050]** El equipo JCT-VC está trabajando en el desarrollo de la norma HEVC. Los esfuerzos de normalización de la HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de la HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a dispositivos existentes de acuerdo, por ejemplo, a la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modalidades de codificación de intra-predicción, el HM puede proporcionar hasta

treinta y tres modalidades de codificación de intra-predicción. Aunque esta divulgación a veces puede basarse en el documento WD7 de la HEVC con fines de explicación, las técnicas de esta divulgación no están de ninguna manera limitadas al documento WD7. El estándar HEVC continúa evolucionando, y se contempla que las técnicas de esta divulgación puedan ser compatibles con versiones futuras de la HEVC. Por ejemplo, las técnicas en esta divulgación pueden ser usadas conjuntamente con el "Borrador 10 de HEVC" o "WD10", que se describe en el documento JCTVC-L1003\_v18, de Bross y col., titulado "High Efficiency vídeo Coding (HEVC) Text Specification Draft 10" ["Especificación textual de la Codificación de Vídeo de Alta Eficacia (HEVC), Borrador 10"], Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, duodécima conferencia: Ginebra, Suiza, 14 al 23 de enero de 2013, que, a partir del 5 de febrero de 2013, puede descargarse desde [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v18.zip](http://phenix.it-sudparis.eu/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v18.zip).

**[0051]** En general, el modelo de funcionamiento del HM describe que una trama o imagen de vídeo puede dividirse en una secuencia de bloques arbolados o unidades de codificación de máximo tamaño (LCU), que incluyen muestras tanto de luma como de croma. Los datos sintácticos dentro de un flujo de bits pueden definir un tamaño para la LCU, que es la máxima unidad de codificación en lo que respecta al número de píxeles. Un fragmento incluye un cierto número de bloques arbolados consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque arbolado puede dividirse en unidades de codificación (CU) de acuerdo a un árbol cuádruple. En general, una estructura de datos de árbol cuádruple incluye un nodo por CU, con un nodo raíz correspondiente al bloque arbolado. 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.

**[0052]** Cada nodo de la estructura de datos de árbol cuádruple puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo en el árbol cuádruple puede incluir un indicador de división, que indica si la CU correspondiente al nodo está dividida o no en varias sub-CU. Los elementos sintácticos para una CU pueden definirse de manera recursiva y pueden depender de si la CU está dividida o no en varias sub-CU. Si una CU no está dividida adicionalmente, se denomina CU hoja. En esta divulgación, cuatro sub-CU de una CU hoja también se denominarán CU hojas incluso aunque no haya una división explícita de la CU hoja original. Por ejemplo, si una CU con un tamaño de 16x16 no se divide adicionalmente, las cuatro sub-CU de tamaño 8x8 también se denominarán CU hojas aunque la CU de tamaño 16x16 no se haya dividido nunca.

**[0053]** Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto que una CU no tiene una distinción de tamaño. Por ejemplo, un bloque arbolado puede dividirse en cuatro nodos secundarios (también denominados sub-CU) y cada nodo secundario puede a su vez ser un nodo principal y dividirse en otros cuatro nodos secundarios. Un nodo secundario final, no dividido, denominado un nodo hoja del árbol cuádruple, comprende un nodo de codificación, también denominado CU hoja. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces que puede dividirse un bloque arbolado, denominado profundidad de CU máxima, y también pueden definir un tamaño mínimo de los nodos de codificación. Por consiguiente, un flujo de bits también puede definir una unidad de codificación mínima (SCU). Esta divulgación utiliza el término "bloque" para referirse a cualquiera entre una CU, PU o TU, en el contexto de la HEVC, o a estructuras de datos similares en el contexto de otras normas (por ejemplo, macrobloques y subbloques de los mismos en la norma H.264/AVC).

**[0054]** Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe ser de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Las modalidades de división pueden diferir en función de si la CU está codificada en modalidad de salto o directa, codificada en modalidad de intra-predicción o codificada en modalidad de inter-predicción. Las PU pueden dividirse para tener forma no cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo a un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

**[0055]** La norma HEVC admite transformaciones de acuerdo a las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU típicamente se basa en el tamaño de las PU de una CU dada, definida para una LCU dividida, aunque puede que no siempre sea así. Las TU son habitualmente del mismo tamaño o de un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas mediante una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformación (TU). Los valores de diferencias de píxeles, asociados a las TU, pueden transformarse para generar coeficientes de transformación, que pueden cuantizarse.

**[0056]** Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa una zona espacial correspondiente a la totalidad, o a una parte, de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. Además, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada en intra-modalidad, pueden incluirse datos para la PU en un árbol cuádruple residual (RQT), que pueden incluir datos que describen una modalidad de intra-predicción para una TU

correspondiente a la PU. Como otro ejemplo, cuando la PU está codificada en la inter-modalidad, la PU puede incluir datos que definen uno o más vectores de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden 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, la Lista 0, la Lista 1 o la Lista C) para el vector de movimiento.

**[0057]** Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Las unidades de transformación pueden especificarse usando un RQT (también denominado estructura de árbol cuádruple de TU), como se ha expuesto anteriormente. Por ejemplo, un indicador de división puede indicar si una CU hoja está dividida en cuatro unidades de transformación. A continuación, cada unidad de transformación puede dividirse adicionalmente en más sub-TU. Cuando una TU no se divide adicionalmente, puede denominarse una TU hoja. En general, en lo que respecta a la intra-codificación, todas las TU hoja que pertenecen a una CU hoja comparten la misma modalidad de intra-predicción. Es decir, la misma modalidad de intra-predicción se aplica en general para calcular valores predichos para todas las TU de una CU hoja. En lo que respecta a la intra-codificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja usando la modalidad de intra-predicción, como una diferencia entre la parte de la CU correspondiente a la TU y el bloque original. Una TU no está necesariamente limitada al tamaño de una PU. De este modo, las TU pueden ser mayores o menores que una PU. En lo que respecta a la intra-codificación, una PU puede estar co-ubicada 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.

**[0058]** Además, las TU de las CU hojas también pueden asociarse a las respectivas estructuras de datos de árbol cuádruple, denominadas árboles cuádruples residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuádruple que indica cómo la CU hoja está dividida en varias TU. El nodo raíz de un árbol cuádruple de TU corresponde en general a una CU hoja, mientras que el nodo raíz de un árbol cuádruple de CU corresponde en general a un bloque arbolado (o LCU). Las TU del RQT que no están divididas se denominan TU hojas. En general, esta divulgación usa los términos CU y TU para hacer referencia a una CU hoja y a una TU hoja, respectivamente, a no ser que se indique lo contrario.

**[0059]** Una secuencia de vídeo incluye habitualmente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, en una cabecera de una o más de las imágenes o en otras ubicaciones, que describen un cierto número de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen una modalidad de codificación para el fragmento respectivo. Un codificador de vídeo 20 actúa habitualmente sobre bloques de vídeo dentro de fragmentos de vídeo individuales con el fin de 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 presentar tamaños fijos o variables y pueden diferir en tamaño de acuerdo a una norma de codificación especificada.

**[0060]** Como ejemplo, el HM presta soporte a la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular sea  $2N \times 2N$ , el HM admite la intra-predicción en tamaños de PU de  $2N \times 2N$  o  $N \times N$  y la inter-predicción en tamaños de PU simétricos de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$  o  $N \times N$ . El HM también presta soporte a la división asimétrica para la inter-predicción en tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$ . En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida en el 25 % y el 75 %. La parte de la CU correspondiente a la división del 25 % está indicada por una "n" seguida de una indicación de "arriba", "abajo", "izquierda" o "derecha". Así, por ejemplo, "2NxnU" se refiere a una CU de tamaño  $2N \times 2N$  que está dividida horizontalmente con una PU de tamaño  $2N \times 0,5N$  encima y una PU de tamaño  $2N \times 1,5N$  debajo.

**[0061]** En esta divulgación, "Nxn" y "N por N" pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo,  $16 \times 16$  píxeles o 16 por 16 píxeles. En general, un bloque de tamaño  $16 \times 16$  tendrá 16 píxeles en la dirección vertical ( $y = 16$ ) y 16 píxeles en la dirección horizontal ( $x = 16$ ). Asimismo, un bloque de tamaño  $N \times N$  presenta, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles en un bloque pueden estar ordenados en filas y columnas. Además, no es necesario que los bloques presenten necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender  $N \times M$  píxeles, donde M no es necesariamente igual a N.

**[0062]** Tras la codificación de intra-predicción o inter-predicción mediante 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 datos sintácticos que describen un procedimiento o modalidad de generación de datos de píxeles predictivos en el dominio espacial (también denominado el dominio de píxeles) y las TU pueden comprender coeficientes en el dominio de transformación, tras la aplicación de una transformación, por ejemplo, una transformación de coseno discreta (DCT), una transformación entera, una transformación de ondículas o una transformación conceptualmente similar, a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y los valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar

las TU incluyendo los datos residuales para la CU, y a continuación transformar las TU para generar coeficientes de transformación para la CU.

5 **[0063]** Tras cualquier transformación para generar coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantización de los coeficientes de transformación. La cuantización se refiere, en general, a un proceso en el que los coeficientes de transformación se cuantizan para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos o a la totalidad de los coeficientes. Por ejemplo, un valor de  $n$  bits puede redondearse por lo bajo a un valor de  $m$  bits durante la cuantización, donde  $n$  es mayor que  $m$ .

10 **[0064]** Después de la cuantización, el codificador de vídeo puede recorrer los coeficientes de transformación, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantizados. El recorrido puede estar diseñado para colocar los coeficientes de energía más alta (y por lo tanto de menor frecuencia) en la parte frontal de la matriz y para colocar los coeficientes de energía más bajos (y por lo tanto de mayor frecuencia) en la parte posterior de la matriz. En algunos ejemplos, el codificador de vídeo 20 puede usar un orden de recorrido predefinido para recorrer los coeficientes de transformación cuantizados, para producir un vector en serie que pueda ser codificado por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneo adaptativo. Después de recorrer los coeficientes de transformación cuantizados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo a la codificación de longitud variable adaptativa de acuerdo al contexto (CAVLC), la codificación aritmética binaria adaptativa de acuerdo al contexto (CABAC), la codificación aritmética binaria adaptativa de acuerdo al contexto basada en la sintaxis (SBAC), la codificación por entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos de codificación por entropía. El codificador de vídeo 20 también puede realizar la codificación por entropía de los elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

20 **[0065]** Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo contextual a un símbolo que se va a transmitir. El contexto puede referirse, por ejemplo, a si los valores contiguos del símbolo son distintos de cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo que se va a transmitir. Las palabras de código en la VLC pueden construirse de forma que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de la VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad puede basarse en un contexto asignado al símbolo.

30 **[0066]** De acuerdo a las técnicas de esta divulgación, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar configurados para ejecutar una función ejemplar "GetLSB (poc, len)", donde poc es un entero y len es un entero positivo, de la siguiente manera.

35

$$\text{GetLSB (poc, len)} = \text{poc} - (1 \ll \text{len}) * \text{Floor} \left( \frac{\text{poc}}{1 \ll \text{len}} \right) \quad (1)$$

40

En el ejemplo de la función (1), "<<" representa un operador de desplazamiento a la izquierda bit a bit y la función Floor() devuelve un valor del argumento pasado a la misma que se redondea hacia abajo.

45 **[0067]** El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para realizar una o más técnicas diferentes para la obtención de conjuntos de imágenes de referencia (RPS) y la señalización de imágenes de referencia a largo plazo (LTRP) para ser incluidas en el RPS de una imagen codificada. En general, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para realizar uno cualquiera o todos los ejemplos descritos en esta divulgación, solos o en cualquier combinación. En general, esta divulgación describe técnicas relacionadas con las siguientes técnicas de codificación y decodificación, donde el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para realizar una cualquiera o todas estas técnicas en combinación con otras técnicas del documento WD7 u otras estructuras de codificación de vídeo.

50 **[0068]** En un ejemplo, la información señalizada para que una LTRP particular sea incluida en el RPS de la imagen actual puede ser el valor de delta POC, ya sea entre la imagen actual y la LTRP o entre la LTRP y la imagen de RAP anterior en el orden de decodificación. Como alternativa, la señalización y obtención de la LTRP pueden permanecer igual que en el documento WD7, excepto porque el elemento sintáctico delta\_poc\_msb\_cycle\_lt [i] siempre se señala para cada LTRP en la cabecera de fragmento, bien eliminando el elemento sintáctico delta\_poc\_msb\_present\_flag [i], o bien ordenando que delta\_poc\_msb\_present\_flag [i] sea 1 para todas las LTRP. Como otra alternativa, la información señalizada para una LTRP particular a incluir en el RPS de la imagen actual puede ser el valor del delta POC entre la LTRP y la imagen de RAP anterior en orden de decodificación, o puede aplicarse el procedimiento alternativo anterior.

55

60

**[0069]** En una alternativa, las imágenes a continuación de una imagen de CRA o BLA en el orden de decodificación pueden ser desautorizadas para utilizar imágenes, anteriores a las imágenes de CRA o BLA en el orden de decodificación, como las LTRP. En consecuencia, las imágenes principales pueden ser desautorizadas para hacer referencia a las imágenes que preceden a la imagen de CRA o BLA asociada, en orden de decodificación, como las LTRP. Para imágenes que no son TFD, el número de bits necesarios para señalar el LSB de la LTRP puede ser determinado por la imagen con el POC más pequeño que no esté marcada como TFD, y que preceda, en orden de decodificación, a la imagen actual y que siga, en el orden de decodificación, a la imagen de RAP anterior en orden de decodificación, incluida la imagen de RAP anterior en orden de decodificación. Para imágenes TFD, el número de bits necesarios para señalar el LSB de la LTRP puede ser determinado por la imagen con el POC más pequeño que preceda, en orden de decodificación, a la imagen actual y que siga, en orden de decodificación, a la imagen de RAP anterior en orden de decodificación, incluida la imagen de RAP anterior en orden de decodificación.

**[0070]** En una alternativa, las imágenes TFD asociadas a una imagen de RAP pueden ser desautorizadas para seguir, en orden de decodificación, a cualquier imagen DLP asociada a la misma imagen de RAP. Para imágenes que no son TFD, el número de bits necesarios para señalar los LSB de la LTRP puede ser determinado por la imagen con el POC más pequeño que no esté marcada como una imagen TFD, y que preceda a la imagen actual, en orden de decodificación, y que siga, en orden de decodificación, a la imagen de RAP anterior en orden de decodificación, incluida la imagen de RAP anterior en orden de decodificación. Para las imágenes TFD, el número de bits necesarios para señalar los LSB de la LTRP puede ser determinado por la imagen con el POC más pequeño que preceda, en orden de decodificación, a la imagen actual y que siga, en orden de salida, al RAP antes de la imagen de RAP anterior en orden de decodificación, incluida la imagen de RAP antes de la imagen de RAP anterior en orden de decodificación.

**[0071]** En una alternativa, las imágenes principales asociadas a una imagen de CRA o BLA pueden ser desautorizadas para referirse a la imagen de CRA o BLA asociada, o a otras imágenes principales asociados a la misma imagen de CRA o BLA, como las LTRP. El número de bits necesarios para señalar el LSB de la LTRP puede ser determinado por la imagen de RAP que precede, en orden de salida, a la imagen actual.

**[0072]** En una alternativa, puede ser impuesta la siguiente restricción. Cualquier imagen que preceda a una imagen de CRA o BLA en orden de decodificación deberá preceder a cualquier imagen DLP asociada a la imagen de CRA o BLA en orden de salida. Además, también puede imponerse la siguiente restricción en cuanto a que cualquier imagen TFD asociada a una imagen de CRA o BLA precederá a cualquier imagen DLP asociada a la imagen de CRA o BLA en orden de salida. Estas restricciones ejemplares no permiten casos en los que las imágenes DLP están intercaladas, en orden de salida, con imágenes TFD o imágenes anteriores, en orden de decodificación, a la imagen de CRA o BLA asociada. En esos casos, cuando se accede aleatoriamente o se realiza una conmutación de flujo desde la imagen de CRA o BLA, la velocidad inicial de imágenes sería menor que después de que se emitan las imágenes DLP. En el flujo de transmisión de adaptación con frecuente conmutación de flujo, por ejemplo, en el contexto del flujo de transmisión dinámico adaptativo por HTTP (DASH), dicha fluctuación en la velocidad de imágenes puede causar una experiencia de usuario indeseable. Para una imagen de CRA o BLA, el documento WD7 proveyó que no habrá ninguna imagen anterior a la imagen de CRA o BLA en orden de decodificación y a continuación de la imagen de CRA o BLA en orden de salida. La inclusión de esa restricción fue por una razón similar.

**[0073]** En una alternativa, las imágenes principales (más específicamente, imágenes TFD) asociadas a una primera imagen son autorizadas a usar las imágenes DLP asociadas a una segunda imagen para referencia de inter-predicción, como las STRP o LTRP, en donde la primera imagen es una imagen de CRA, la segunda imagen es una imagen de CRA o BLA y la segunda imagen y las primeras imágenes son dos imágenes de RAP continuas en orden de decodificación, con la primera imagen a continuación de la segunda imagen en orden de decodificación. Tal referencia de inter-predicción de imágenes DLP por parte de imágenes principales puede habilitarse permitiendo que tales imágenes DLP estén presentes en el conjunto de imágenes de referencia de dichas imágenes principales, y puede permitir una codificación más flexible y de mayor eficacia de las imágenes principales. En consecuencia, las definiciones de imágenes de CRA y BLA se pueden cambiar para permitir tal inter-predicción.

**[0074]** En una alternativa, puede ser desautorizado el entrelazado en orden de decodificación de las imágenes TFD asociadas a una imagen de CRA o BLA y de las imágenes a continuación de la imagen de CRA o BLA, tanto en orden de decodificación como en orden de salida, es decir, imágenes de cola de la imagen de CRA o BLA. En otras palabras, puede requerirse que cualquier imagen TFD asociada a una imagen de CRA o BLA preceda a cualquier imagen de cola de la imagen de CRA o BLA, en orden de decodificación.

**[0075]** Como alternativa, puede ser desautorizado el entrelazado en orden de decodificación de las imágenes principales asociadas a una imagen de CRA o BLA y de las imágenes a continuación de la imagen de CRA o BLA, tanto en orden de decodificación como en orden de salida. En otras palabras, puede requerirse que cualquier imagen principal asociada a una imagen de CRA o BLA preceda a cualquier imagen, en orden de decodificación, que siga a la imagen de CRA o BLA, tanto en orden de decodificación como en orden de salida.

**[0076]** Los siguientes ejemplos de implementaciones proporcionan más detalle por los ejemplos descritos

anteriormente. Debería entenderse que las diversas técnicas descritas en las implementaciones ejemplares pueden usarse solas o en cualquier combinación.

5 **[0077]** Un primer ejemplo de implementación proporciona un procedimiento para intentar resolver el problema de la resistencia de errores y para asegurarse de que la desviación del RPS sea autocontenida (es decir, no dependiente del estado de la DPB), y esté al mismo tiempo libre de problemas cuando se realiza un acceso aleatorio desde una imagen de CRA, o cuando una imagen de CRA se convierte en una imagen de BLA. En esta primera implementación ejemplar, la información señalizada para una LTRP particular, a incluir en el RPS de la imagen actual, es el valor de delta POC, ya sea entre la imagen actual y la LTRP (modalidad 1) o entre la LTRP y la imagen de RAP anterior en orden de decodificación (modalidad 2). Esta primera implementación ejemplar también está alineada con la señalización y la obtención del RPS actual para las STRP del documento WD7. Los codificadores, tales como el codificador de vídeo 20, pueden elegir una de las dos modalidades para cada LTRP a señalizar, la que daría como resultado la utilización de menos bits. Por ejemplo, si la LTRP está más cerca del RAP anterior en orden de decodificación que la imagen actual, en la distancia de orden de salida / visualización, entonces el codificador de vídeo 20 puede configurarse para usar la modalidad 2. De lo contrario, el codificador de vídeo 20 puede usar la modalidad 1. Asimismo, el decodificador 30 de vídeo puede recibir datos sintácticos que describen la modalidad a usar, y usar la modalidad señalizada por los datos sintácticos.

20 **[0078]** Para lograr el primer ejemplo de implementación, pueden ser utilizados los siguientes cambios de sintaxis, semántica y proceso de codificación, en relación con el documento WD7, donde el texto subrayado representa un cambio en relación con el documento WD7. La Tabla 5 a continuación proporciona un conjunto ejemplar de sintaxis para un conjunto de parámetros de imagen (PPS).

**TABLA 5**

	<b>Descriptor</b>
<code>pic_parameter_set_rbsp() {</code>	
<code>  pic_parameter_set_id</code>	<code>ue(v)</code>
<code>  ...</code>	
<code>  num_ref_idx_10_default_active_minus1</code>	<code>ue(v)</code>
<code>  num_ref_idx_l1_default_active_minus1</code>	<code>ue(v)</code>
<code>  poc_lt_idc</code>	<code>u(2)</code>
<code>  ...</code>	
<code>  rbsp_trailing_bits()</code>	
<code>}</code>	

25 **[0079]** En este primer ejemplo de implementación, como se muestra en la Tabla 5, la sintaxis del PPS incluye un elemento sintáctico adicional, `poc_lt_idc`, en relación con el documento WD7. La semántica para otros elementos sintácticos puede seguir siendo la misma. En este ejemplo, `poc_lt_idc` se puede usar para indicar la presencia del `delta_poc_lt_curr_pic_flag[i]` en la cabecera de fragmento y, cuando el `delta_poc_lt_curr_pic_flag[i]` no está presente en la cabecera de fragmento, para determinar el valor deducido del `delta_poc_lt_curr_pic_flag[i]` en la cabecera de fragmento. Si `poc_lt_idc` es igual a 0 o 1, `delta_poc_lt_curr_pic_flag[i]` no está presente en la cabecera de fragmento y se deduce que el valor es igual a `poc_lt_idc`, en este ejemplo. Cuando `poc_lt_idc` es igual a 2, `delta_poc_lt_curr_pic_flag[i]` se señala en la cabecera de fragmento, en este ejemplo. El valor 3 para `poc_lt_idc` está reservado para uso futuro, en este ejemplo.

35 **[0080]** La Tabla 6 proporciona un conjunto ejemplar de sintaxis para una cabecera de fragmento de acuerdo a la primera implementación ejemplar.

**TABLA 6**

	<b>Descriptor</b>
<code>slice_header() {</code>	
<code>  ...</code>	
<code>  si (long_term_ref_pics_present_flag) {</code>	
<code>    num_long_term_pics</code>	<code>ue(v)</code>
<code>    para( i = 0; i &lt; num_long_term_pics; i++ ) {</code>	
<code>      si (poc_lt_idc == 2)</code>	
<code>        delta_poc_lt_curr_pic_flag [i]</code>	<code>u(1)</code>
<code>        long_delta_poc_lp [i]</code>	<code>ue(v)</code>
<code>      si (long_delta_poc_lt [i] != 0)</code>	
<code>        delta_poc_lt [i]</code>	<code>u(v)</code>
<code>        used_by_curr_pic_lt_flag [i]</code>	<code>u(1)</code>
<code>    }</code>	
<code>  }</code>	
<code>  ...</code>	
<code>}</code>	

40 **[0081]** En este ejemplo, la cabecera de fragmento incluye elementos sintácticos adicionales de

delta\_poc\_lt\_curr\_pic\_flag [i], delta\_poc\_lt\_len [i] y, en algunos casos, delta\_poc\_lt [i]. A continuación, se describe un conjunto ejemplar de reglas semánticas para estos elementos sintácticos. Las reglas semánticas para otros elementos sintácticos pueden quedar igual que en el documento WD7.

5 **[0082]** En esta primera implementación ejemplar, indicador\_imagen\_actual\_delta\_poc\_lp [i] igual a 1 puede especificar que delta\_poc\_lp [i] representa la diferencia de recuento de orden de imagen entre la imagen actual y la i-ésima imagen de referencia a largo plazo. indicador\_imagen\_actual\_delta\_poc\_lp [i] igual a 0 puede especificar que delta\_poc\_lp [i] representa la diferencia de recuento de orden de imagen entre la imagen de RAP anterior en orden de decodificación y la i-ésima imagen de referencia a largo plazo. Si poc\_lt\_idc (por ejemplo, de la Tabla 5) es igual a 0 o 1, se puede deducir que el valor de delta\_poc\_lt\_curr\_pic\_flag [i] es igual a poc\_lt\_idc.

10 **[0083]** En esta primera implementación ejemplar, delta\_poc\_lt\_len [i] se puede utilizar para especificar el número de bits para representar delta\_poc\_lt [i]. Cuando delta\_poc\_lt\_curr\_pic\_flag [i] es igual a 0, el valor de delta\_poc\_lt\_len [i] puede estar en el intervalo entre 0 y 24, inclusive. En caso contrario (delta\_poc\_lt\_curr\_pic\_flag [i] es igual a 1), el valor de delta\_poc\_lt\_len [i] puede estar en el intervalo entre 1 y 24, inclusive.

15 **[0084]** La variable DeltaPocLtLen [i] puede obtenerse como se indica a continuación:

20 si (i = 0)  
DeltaPocLtLen [i] = delta\_poc\_lt\_len [i]  
en caso contrario  
DeltaPocLtLen [i] = DeltaPocLtLen [i - 1] + delta\_poc\_lt\_len [i]

25 **[0085]** Como alternativa, delta\_poc\_lt\_len [i] se puede cambiar por poc\_lt\_len [i], que se codifica con longitud fija, por ejemplo, como u (N), siendo N igual a 5. En este caso, la semántica puede ser la siguiente. poc\_lt\_len [i] puede especificar el número de bits para representar delta\_poc\_lt [i]. Si delta\_poc\_lt\_curr\_pic\_flag [i] es igual a 0, el valor de poc\_lt\_len [i] puede estar en el rango de 0 a 24, inclusive. De lo contrario, (delta\_poc\_lt\_curr\_pic\_flag [i] es igual a 1), el valor de poc\_lt\_len [i] puede estar en el rango de 1 a 24, inclusive. La variable DeltaPocLtLen [i] puede configurarse para que sea igual a poc\_lt\_len [i].

30 **[0086]** En esta primera implementación ejemplar, delta\_poc\_lt [i] puede especificar el valor del delta de recuento de orden de imágenes de la i-ésima imagen de referencia a largo plazo que se incluye en el conjunto de imágenes de referencia a largo plazo de la imagen actual. La longitud de delta\_poc\_lt [i] puede ser DeltaPocLtLen [i].

35 **[0087]** El decodificador de vídeo 30 se puede configurar para realizar un proceso de decodificación para un conjunto de imágenes de referencia de acuerdo a esta primera implementación ejemplar, de la siguiente manera.

40 **[0088]** El codificador de vídeo 20 puede realizar un proceso similar mientras codifica datos de vídeo para generar el conjunto de imágenes de referencia. El decodificador de vídeo 30 puede invocar este proceso de decodificación una vez por imagen, después de la decodificación de una cabecera de fragmento, pero antes de la decodificación de cualquier unidad de codificación y antes del proceso de decodificación para la construcción de la lista de imágenes de referencia del fragmento, como se especifica en la subcláusula 8.3.3 del documento WD7. El proceso puede dar como resultado marcar una o más imágenes de referencia como "no usada como referencia". Las citas del documento WD7 que comienzan con "NOTA" deberían reconocerse simplemente como citas tomadas directamente del documento WD7 y no como necesarias para las técnicas de esta divulgación.

50 NOTA 1 - El conjunto de imágenes de referencia es una descripción absoluta de las imágenes de referencia utilizadas en el proceso de decodificación de las imágenes codificadas actuales y futuras. La señalización del conjunto de imágenes de referencia es explícita en el sentido de que todas las imágenes de referencia incluidas en el conjunto de imágenes de referencia se enumeran explícitamente y que no existe ningún proceso predeterminado de construcción del conjunto de imágenes de referencia en el decodificador que dependa del estado de la memoria intermedia de imágenes decodificadas.

55 **[0089]** Las imágenes de referencia pueden ser identificados por sus valores PicOrderCntVal.

60 **[0090]** Cinco listas de valores de recuento de orden de imagen pueden ser construidas para obtener el conjunto de imágenes de referencia: PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr y PocLtFoll, con un número de elementos igual a NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr y NumPocLtFoll, respectivamente.

65 **[0091]** Si la imagen actual es una imagen de CRA que es la primera imagen codificada en el flujo de bits, una imagen de IDR o una imagen de BLA, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr y PocLtFoll se configuran como vacíos, y NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr y NumPocLtFoll se pueden fijar en 0.

**[0092]** De lo contrario, puede aplicarse lo siguiente para la obtención de las cinco listas de valores de recuento de



orden de imágenes y del número de entradas.

```

para (i = 0, j = 0, k = 0; i < NumNegativePics [StRpsIdx]; i++)
  si (UsedByCurrPicS0 [StRpsIdx] [i])
5     PocStCurrBefore [j++] = PicOrderCntVal + DeltaPocS0 [StRpsIdx] [i]
    en caso contrario
      PocStFoll [k++] = PicOrderCntVal + DeltaPocS0 [StRpsIdx] [i]
  NumPocStCurrBefore = j
para (i = 0, j = 0; i < NumPositivePics [StRpsIdx]; i++)
10     si (UsedByCurrPicS1 [StRpsIdx] [i])
      PocStCurrAfter [j++] = PicOrderCntVal + DeltaPocS1 [StRpsIdx] [i]
    en caso contrario
      PocStFoll [k++] = PicOrderCntVal + DeltaPocS1 [StRpsIdx] [i]
  NumPocStCurrAfter = j
15  NumPocStFoll = k (8-5)
para (i = 0, j = 0, k = 0; i < num_long_term_pics; i++)
  si (used_by_curr_pic_lt_flag [i]) {
    si (delta_poc_lt_curr_pic_flag [i])
20     PocLtCurr [j] = PicOrderCntVal - delta_poc_lt [i]
    en caso contrario
      PocLtCurr [j] = PrevRapPicPoc + delta_poc_lt [i]
    j++
  }
  en caso contrario {
25     si (delta_poc_lt_curr_pic_flag [i])
      PocLtFoll [k] = PicOrderCntVal - delta_poc_lt [i]
    en caso contrario
      PocLtFoll [k] = PrevRapPicPoc + delta_poc_lt [i]
    k++
30  }
  NumPocLtCurr = j
  NumPocLtFoll = k

```

35 donde PicOrderCntVal y PrevRapPicPoc son los valores de recuento de orden de imagen de la imagen actual y de la imagen RAP anterior en orden de decodificación, respectivamente, como se especifica en la subcláusula 8.2.1.

40 NOTA 2 - Un valor de StRpsIdx en el rango de 0 a num\_short\_term\_ref\_pic\_sets - 1, inclusive, puede indicar que se está utilizando un conjunto de imágenes de referencia a corto plazo del conjunto de parámetros de secuencia activa, donde StRpsIdx es el índice del conjunto de imágenes de referencia a corto plazo para la lista de conjuntos de imágenes de referencia a corto plazo en el orden en que se señalizan en el conjunto de parámetros de secuencia. StRpsIdx igual a num\_short\_term\_ref\_pic\_sets indica que se está utilizando un conjunto de imágenes de referencia a corto plazo explícitamente señalado en la cabecera de fragmento.

45 **[0093]** El conjunto de imágenes de referencia puede incluir cinco listas de imágenes de referencia: RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetStFoll, RefPicSetLtCurr y RefPicSetLtFoll. La variable NumPocTotalCurr se puede fijar igual a NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr. Al decodificar un fragmento P o B, el documento WD7 requiere, para la conformidad del flujo de bits, que el valor de NumPocTotalCurr no sea igual a 0.

50 NOTA 3 - El RefPicSetStCurrBefore, el RefPicSetStCurrAfter y el RefPicSetLtCurr pueden contener todas las imágenes de referencia que pueden usarse en la inter-predicción de la imagen actual y que pueden usarse en la inter-predicción de una o más de las imágenes que siguen a la imagen actual en orden de decodificación. El RefPicSetStFoll y el RefPicSetLtFoll pueden incluir todas las imágenes de referencia que no se utilizan en la inter-predicción de la imagen actual, pero se pueden usar en la inter-predicción de una o más de las imágenes siguientes a la imagen actual en orden de decodificación.

60 **[0094]** El marcado de una imagen de referencia puede ser "no utilizada para referencia", "utilizada como referencia de corto plazo," o "utilizada para referencia a largo plazo", pero sólo uno entre estos tres, en el documento WD7. Cuando se hace referencia a una imagen de referencia marcada como "utilizada como referencia", esto se refiere colectivamente a la imagen marcada como "usada para referencia a corto plazo" o "utilizada para referencia a largo plazo" (pero no ambas en el documento WD7) . Una imagen de referencia que está marcada como "utilizada para referencia a corto plazo" se denomina imagen de referencia a corto plazo. Una imagen de referencia que está marcada como "utilizada para referencia a largo plazo" se denomina imagen de referencia a largo plazo.

65 **[0095]** El proceso de obtención para el conjunto de imágenes de referencia y el marcado de imágenes se llevan a cabo de acuerdo a las siguientes etapas ordenadas, donde DPB se refiere a la memoria intermedia de imágenes

decodificadas, como se describe en el Anexo C del documento WD7, donde (# - #) se refiere a un número de ecuación del documento WD7:

1. Vale lo siguiente:

```

5      para( i = 0; i < NumPocLtCurr; i++ ) {
          si (hay una imagen imagenX en la DPB con PicOrderCntVal igual a PocLtCurr [i])
              RefPicSetLtCurr [i] = imagenX
          en caso contrario
10         RefPicSetLtCurr [i] = "sin imagen de referencia"
        } (8-6)
        para( i = 0; i < NumPocLtFoll; i++ ) {
          si (hay una imagen imagenX en la DPB con PicOrderCntVal igual a PocLtFoll [i])
              RefPicSetLtFoll [i] = imagenX
15         en caso contrario
              RefPicSetLtFoll [i] = "sin imagen de referencia"
        }
    
```

2. Todas las imágenes de referencia incluidas en el RefPicSetLtCurr y el RefPicSetLtFoll están marcadas como "utilizada para referencia a largo plazo"

3. Vale lo siguiente:

```

25     para(i = 0; i < NumPocStCurrBefore; i++)
          si (hay una imagen imagenX en la DPB con PicOrderCntVal igual a PocStCurrBefore [i])
              RefPicSetStCurrBefore [i] = imagenX

          en caso contrario

30         RefPicSetStCurrBefore [i] = "sin imagen de referencia"

        para(i = 0; i < NumPocStCurrAfter; i++)
          si (hay una imagen imagenX en la DPB con PicOrderCntVal igual a PocStCurrAfter [i])
              RefPicSetStCurrAfter [i] = imagenX
35         en caso contrario

              RefPicSetStCurrAfter [i] = "sin imagen de referencia" (8-7)

40         para(i = 0; i < NumPocStFoll; i++)
          si (hay un imagenX en la DPB con PicOrderCntVal igual a PocStFoll [i])
              RefPicSetStFoll [i] = imagenX

          en caso contrario

45         RefPicSetStFoll [i] = "sin imagen de referencia"
    
```

4. Todas las imágenes de referencia incluidas en RefPicSetStCurrBefore, RefPicSetStCurrAfter y RefPicSetStFoll están marcadas como "utilizada para referencia a corto plazo".

**[0096]** Todas las imágenes de referencia en la memoria intermedia de imágenes decodificadas que no están incluidos en RefPicSetLtCurr, RefPicSetLtFoll, RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetStFoll se pueden marcar como "no utilizada como referencia".

NOTA 4 - Puede haber una o más imágenes de referencia que están incluidas en el conjunto de imágenes de referencia pero que no están presentes en la memoria intermedia de imágenes decodificadas. Las entradas en RefPicSetStFoll o RefPicSetLtFoll que son iguales a "sin imagen de referencia" deberían ignorarse. A menos que cualquiera de las dos condiciones siguientes sea verdadera, se debe deducir una pérdida de imagen involuntaria, en el documento WD7, para cada entrada en RefPicSetStCurrBefore, RefPicSetStCurrAfter y RefPicSetLtCurr que sea igual a "sin imagen de referencia": a) la primera imagen codificada en el flujo de bits es una imagen de CRA y la imagen codificada actual es una imagen TFD asociada a la primera imagen codificada en el flujo de bits; b) la imagen de RAP anterior que precede a la imagen codificada actual en orden de decodificación es una imagen de BLA y la imagen codificada actual es una imagen TFD asociada a la imagen de BLA.

**[0097]** Como alternativa, el documento WD7 puede exigir que, en lo anterior, cada imagenX identificado en la DPB se marcará como "utilizada como referencia." Alternativamente, el documento WD7 puede restringir que una imagen

marcada como "no utilizada para referencia" nunca se marque como "usada como referencia".

**[0098]** Alternativamente, los cinco pasos anteriores puede ser reemplazados por lo siguiente:

5 1. Vale lo siguiente:

```

para( i = 0; i < NumPocLtCurr; i++ ) {
    si (hay una imagen imagenX en la DPB que tiene PicOrderCntVal igual a PocLtCurr [i] y
    que
    
```

```

10         está marcada como "usada como referencia")
        RefPicSetLtCurr [i] = imagenX
    en caso contrario
    
```

```

15         RefPicSetLtCurr [i] = "sin imagen de referencia"
    }
    
```

(8-6)

```

para( i = 0; i < NumPocLtFoll; i++ ) {
    si (hay una imagen imagenX en la DPB que tiene PicOrderCntVal igual a PocLtFoll [i] y
    que
    
```

```

20         está marcada como "usada como referencia")
        RefPicSetLtFoll [i] = imagenX
    en caso contrario
    
```

```

25         RefPicSetLtFoll [i] = "sin imagen de referencia"
    }
    
```

2. Todas las imágenes de referencia incluidas en el RefPicSetLtCurr y el RefPicSetLtFoll están marcadas como "utilizada para referencia a largo plazo"

30 3. Vale lo siguiente:

```

para(i = 0; i < NumPocStCurrBefore; i++)
    si (hay una imagen imagenX en la DPB que tiene PicOrderCntVal igual a PocStCurrBefore [i]
    y que
    
```

```

35         está marcada como "usada para referencia")
        RefPicSetStCurrBefore [i] = imagenX
    en caso contrario
    
```

```

        RefPicSetStCurrBefore [i] = "sin imagen de referencia"
    
```

```

40 para(i = 0; i < NumPocStCurrAfter; i++)
    si (hay una imagen imagenX en el DPB que tiene PicOrderCntVal igual a PocStCurrAfter [i]
    y que
    
```

```

45         está marcada como "usada para referencia")
        RefPicSetStCurrAfter [i] = imagenX
    en caso contrario
    
```

```

        RefPicSetStCurrAfter [i] = "sin imagen de referencia"
    (8-7)
    
```

```

50 para(i = 0; i < NumPocStFoll; i++)
    si (hay un imagenX en la DPB que tiene PicOrderCntVal igual a PocStFoll [i] y que
        está marcada como "usada para referencia")
        RefPicSetStFoll [i] = imagenX
    
```

```

55         en caso contrario
        RefPicSetStFoll [i] = "sin imagen de referencia"
    
```

4. Todas las imágenes de referencia incluidas en RefPicSetStCurrBefore, RefPicSetStCurrAfter y RefPicSetStFoll están marcadas como "utilizada para referencia a corto plazo".

60 5. Todas las imágenes de referencia en la memoria intermedia de imágenes decodificadas que no están incluidas en RefPicSetLtCurr, RefPicSetLtFoll, RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetStFoll están marcadas como "no utilizada como referencia".

65 NOTA 4 - Puede haber una o más imágenes de referencia que están incluidas en el conjunto de imágenes de referencia pero que no están presentes en la memoria intermedia de imágenes decodificadas. Las entradas en RefPicSetStFoll o RefPicSetLtFoll que son iguales a "sin imagen de referencia" deben ignorarse, de

acuerdo al documento WD7. A menos que cualquiera de las dos condiciones siguientes sea verdadera, se debería deducir una pérdida de imagen involuntaria para cada entrada en RefPicSetStCurrBefore, RefPicSetStCurrAfter y RefPicSetLtCurr que sea igual a "sin imagen de referencia" en el documento WD7: a) la primera imagen codificada en el flujo de bits es una imagen de CRA y la imagen codificada actual es una imagen TFD asociada a la primera imagen codificada en el flujo de bits; b) la imagen de RAP anterior que precede a la imagen codificada actual en orden de decodificación es una imagen de BLA y la imagen codificada actual es una imagen TFD asociada a la imagen de BLA.

**[0099]** Como alternativa, el orden de las etapas anteriores se puede cambiar a 3, 4, 1, 2 y 5.

**[0100]** En algunos ejemplos, el documento WD7 requiere, para la conformidad del flujo de bits, que el conjunto de imágenes de referencia esté restringido de la siguiente manera:

- No habrá ninguna imagen de referencia con Id\_temporal mayor que el de la imagen actual incluida en RefPicSetStCurrBefore, RefPicSetStCurrAfter y RefPicSetLtCurr.
- Cuando la imagen actual es una imagen de TLA, no debería haber ninguna imagen de referencia incluida en el conjunto de imágenes de referencia con Id\_temporal mayor o igual que el Id\_temporal de la imagen actual.
- No se incluirá ninguna imagen de referencia en el conjunto de imágenes de referencia que preceda, en orden de salida, a cualquier imagen de RAP que preceda a la imagen actual, tanto en orden de decodificación como en orden de salida.
- A menos que cualquiera de las siguientes condiciones sea verdadera, no habrá ninguna entrada en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetLtCurr que sea igual a "sin imagen de referencia": a) la primera imagen codificada en el flujo de bits es una imagen de CRA y la imagen codificada actual es una imagen TFD asociada a la primera imagen codificada en el flujo de bits; b) la imagen de RAP anterior que precede a la imagen codificada actual en orden de decodificación es una imagen de BLA y la imagen codificada actual es una imagen TFD asociada a la imagen de BLA.

NOTA 5 - Una imagen de referencia no se puede incluir en más de una de las cinco listas de conjuntos de imágenes de referencia.

**[0101]** En algunos ejemplos, el documento WD7 establece que las restricciones anteriores pueden ser las siguientes:

- No habrá ninguna imagen de referencia con Id\_temporal mayor que el de la imagen actual incluida en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetLtCurr.
- Cuando la imagen actual es una imagen de TLA, no debería haber ninguna imagen de referencia incluida en el conjunto de imágenes de referencia con Id\_temporal mayor o igual que el Id\_temporal de la imagen actual.
- Sea prevRapPic la imagen de RAP anterior en orden de decodificación. Cuando la imagen actual es una imagen de RAP o precede, en orden de salida, a prevRapPic, no habrá ninguna imagen de referencia, incluida en el conjunto de imágenes de referencia, que preceda, en orden de salida, a cualquier imagen de RAP que preceda a la imagen actual, tanto en orden de decodificación como en orden de salida.
- Sea prevRapPic la imagen de RAP anterior en orden de decodificación. Cuando la imagen actual no es una imagen de RAP y sigue a prevRapPic en orden de salida, no se incluirá ninguna imagen de referencia en el conjunto de imágenes de referencia que preceda a prevRapPic en orden de decodificación, y no se incluirá ninguna imagen de referencia en el conjunto de imágenes de referencia que sean imágenes TFD.
- A menos que cualquiera de las siguientes condiciones sea verdadera, no habrá ninguna entrada en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetLtCurr que sea igual a "sin imagen de referencia": a) la primera imagen codificada en el flujo de bits es una imagen de CRA y la imagen codificada actual es una imagen TFD asociada a la primera imagen codificada en el flujo de bits; b) la imagen de RAP anterior que precede a la imagen codificada actual en orden de decodificación es una imagen de BLA y la imagen codificada actual es una imagen TFD asociada a la imagen de BLA.

NOTA 5 - Una imagen de referencia no se puede incluir en más de una de las cinco listas de conjuntos de imágenes de referencia, de acuerdo al documento WD7.

**[0102]** Una segunda implementación ejemplar, que se basa en la estructura sintáctica en el documento WD7 de la HEVC, incluye un procedimiento para señalar siempre el ciclo de MSB de la LTRP y eliminar el elemento sintáctico delta\_poc\_msb\_present\_flag [i]. Nuevamente, debería entenderse que varias técnicas de la primera implementación ejemplar y de la segunda implementación ejemplar pueden combinarse en cualquier combinación. El codificador de

vídeo 20 y el decodificador de vídeo 30 pueden configurarse para realizar cualquiera de, o todas, las técnicas de esta segunda implementación ejemplar, sola o en combinación con una o más de las técnicas de la primera implementación ejemplar expuesta anteriormente. La Tabla 7 a continuación proporciona un conjunto ejemplar de sintaxis para una cabecera de fragmento de acuerdo a esta segunda implementación ejemplar.

5

TABLA 7

	Descriptor
slice_header() {	
...	u(l)
<b>short_term_ref_pic_set_idx</b>	u(v)
si (long_term_ref_pics_present_flag) {	
<b>num_long_term_pics</b>	ue(v)
para( i = 0; i < num_long_term_pics; i++ ) {	
<b>poc_lst_lt [i]</b>	u(v)
<b>delta_poc_msb_cycle_lt [i]</b>	ue(v)
<b>used_by_curr_pic_lt_flag [i]</b>	u(l)
}	
}	
...	
byte_alignment ()	
}	

[0103] En la sintaxis ejemplar del encabezado de fragmento de la Tabla 7, se agrega el elemento sintáctico delta\_poc\_msb\_cycle\_lt [i], en relación con el documento WD7. La semántica para este elemento sintáctico puede ser como se describe a continuación, mientras que la semántica para otros elementos sintácticos puede permanecer igual que en el documento WD7, a menos que se indique lo contrario. En esta segunda implementación ejemplar, delta\_poc\_msb\_cycle\_lt [i] se puede usar para determinar el valor de los bits más significativos del valor de recuento de orden de imagen de la i-ésima imagen de referencia a largo plazo que se incluye en el conjunto de imágenes de referencia a largo plazo de la imagen actual.

10

15

[0104] En esta segunda implementación ejemplar, la variable DeltaPocMSBCycleLt [i] puede obtenerse de la siguiente manera, donde (# - #) se refiere a un número de ecuación del documento WD7 y el texto subrayado representa un cambio relativo al documento WD7:

20

```

si (i = 0)
    DeltaPocMSBCycleLt [i] = delta_poc_msb_cycle_lt [i];
en caso contrario (7-37)
    DeltaPocMSBCycleLt [i] = delta_poc_msb_cycle_lt [i] +
        DeltaPocMSBCycleLt [i - 1];
    
```

25

[0105] El decodificador de vídeo 30 puede realizar el siguiente proceso de decodificación para un conjunto de imágenes de referencia. El codificador de vídeo 20 puede configurarse para realizar un proceso esencialmente similar mientras codifica datos de vídeo para generar el conjunto de imágenes de referencia. Este proceso se invoca una vez por imagen, después de la decodificación de una cabecera de fragmento, pero antes de la decodificación de cualquier unidad de codificación y antes del proceso de decodificación para la construcción de la lista de imágenes de referencia del fragmento, como se especifica en la subcláusula 8.3.3 del documento WD7. El proceso puede dar como resultado marcar una o más imágenes de referencia como "no usada como referencia".

30

35

NOTA 1 - El conjunto de imágenes de referencia es una descripción absoluta de las imágenes de referencia utilizadas en el proceso de decodificación de las imágenes codificadas actuales y futuras. La señalización del conjunto de imágenes de referencia es explícita en el sentido de que todas las imágenes de referencia incluidas en el conjunto de imágenes de referencia se enumeran explícitamente y que no existe ningún proceso predeterminado de construcción del conjunto de imágenes de referencia en el decodificador que dependa del estado de la memoria intermedia de imágenes decodificadas.

40

[0106] Las imágenes de referencia se identifican por sus valores de PicOrderCntVal en esta segunda implementación ejemplar.

45

[0107] Cinco listas de valores de recuento de orden de imagen se pueden construir para obtener el conjunto de imágenes de referencia: PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr y PocLtFoll, con números de elementos iguales a NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr y NumPocLtFoll, respectivamente.

50

- Si la imagen actual es una imagen de CRA que es la primera imagen codificada en el flujo de bits, una imagen de IDR o una imagen de BLA, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr y PocLtFoll se configuran como vacíos, y NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr y NumPocLtFoll

están todos fijados en 0.

- De lo contrario, vale lo siguiente para la obtención de las cinco listas de valores de recuento de orden de imágenes y del número de entradas.

```

5      para (i = 0, j = 0, k = 0; i < NumNegativePics [StRpsIdx]; i++)
        si (UsedByCurrPicS0 [StRpsIdx] [i])
            PocStCurrBefore [j++] = PicOrderCntVal + DeltaPocS0 [StRpsIdx] [i]
        en caso contrario
10         PocStFoll [k++] = PicOrderCntVal + DeltaPocS0 [StRpsIdx] [i]
        NumPocStCurrBefore = j
        para (i = 0, j = 0; i < NumPositivePics [StRpsIdx]; i++)
            si (UsedByCurrPicS1 [StRpsIdx] [i])
                PocStCurrAfter [j++] = PicOrderCntVal + DeltaPocS1 [StRpsIdx] [i]
15         en caso contrario
            PocStFoll [k++] = PicOrderCntVal + DeltaPocS1 [StRpsIdx] [i]
        NumPocStCurrAfter = j
        NumPocStFoll = k (8-5)
        para (i = 0, j = 0, k = 0; i < num_long_term_pics; i++)
            si (used_by_curr_pic_lt_flag [i])
                PocLtCurr [j++] = PicOrderCntVal - DeltaPocMSBCycleLt [i] * MaxPicOrderCntLsb - pic_order_cnt_lsb
                + poc_lst_lt [i]
            en caso contrario
20             PocLtFoll [k++] = PicOrderCntVal - DeltaPocMSBCycleLt [i] * MaxPicOrderCntLsb - pic_order_cnt_lsb
                + poc_lst_lt [i]
25             NumPocLtCurr = j
            NumPocLtFoll = k

```

donde PicOrderCntVal es el recuento de orden de imagen de la imagen actual como se especifica en la subcláusula 8.2.1 del documento WD7.

NOTA 2 - Un valor de StRpsIdx en el rango de 0 a num\_short\_term\_ref\_pic\_sets - 1, inclusive, en este ejemplo, indica que se está utilizando un conjunto de imágenes de referencia a corto plazo del conjunto de parámetros de secuencia activa, donde StRpsIdx es el índice del conjunto de imágenes de referencia a corto plazo para la lista de conjuntos de imágenes de referencia a corto plazo en el orden en que se señalizan en el conjunto de parámetros de secuencia. StRpsIdx igual a num\_short\_term\_ref\_pic\_sets indica que se está utilizando un conjunto de imágenes de referencia a corto plazo explícitamente señalado en la cabecera de fragmento.

**[0108]** El conjunto de imágenes de referencia puede incluir cinco listas de imágenes de referencia: RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetStFoll, RefPicSetLtCurr y RefPicSetLtFoll. La variable NumPocTotalCurr se puede fijar igual a NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr. Al decodificar un fragmento P o B, el documento WD7 requiere, para la conformidad del flujo de bits, que el valor de NumPocTotalCurr no sea igual a 0.

NOTA 3 - El RefPicSetStCurrBefore, el RefPicSetStCurrAfter y el RefPicSetLtCurr pueden contener todas las imágenes de referencia que pueden usarse en la inter-predicción de la imagen actual y que pueden usarse en la inter-predicción de una o más de las imágenes que siguen a la imagen actual en orden de decodificación. El RefPicSetStFoll y el RefPicSetLtFoll pueden incluir todas las imágenes de referencia que no se utilizan en la inter-predicción de la imagen actual, pero se pueden usar en la inter-predicción de una o más de las imágenes siguientes a la imagen actual en orden de decodificación.

**[0109]** El marcado de una imagen de referencia puede ser "no usada para referencia", "usada para referencia a corto plazo" o "usada para referencia a largo plazo", pero solo uno de estos tres en el documento WD7. Cuando se hace referencia a una imagen de referencia como marcada como "utilizada como referencia", esta se refiere colectivamente a la imagen marcada como "usada para referencia a corto plazo" o "utilizada para referencia a largo plazo" (pero no ambas, en el documento WD7). Una imagen de referencia que está marcada como "utilizada para referencia a corto plazo" se denomina imagen de referencia a corto plazo. Una imagen de referencia que está marcada como "utilizada para referencia a largo plazo" se denomina imagen de referencia a largo plazo.

**[0110]** El proceso de obtención para el conjunto de imágenes de referencia y el marcado de imágenes se puede realizar, para la segunda implementación ejemplar, de la misma manera como en el primer ejemplo de implementación.

**[0111]** Un tercer ejemplo de implementación proporciona un procedimiento de señalización simplificado de LTRP, junto con una restricción, en cuanto a que las imágenes siguientes a una imagen de CRA o de BLA, en orden de decodificación, son desautorizadas para utilizar imágenes, anteriores a las imágenes de CRA o BLA en el orden de

decodificación, como las LTRP. En consecuencia, las imágenes principales son desautorizadas para referirse a imágenes que preceden a la imagen de CRA o BLA asociada, en orden de decodificación, como las LTRP. Para imágenes que no son TFD, el número de bits necesarios para señalar el LSB de la LTRP puede ser determinado por la imagen con el POC más pequeño que no esté marcada como TFD, y que preceda, en orden de decodificación, a la imagen actual y que siga, en el orden de decodificación, a la imagen de RAP anterior en orden de decodificación, incluida la imagen de RAP anterior en orden de decodificación.

**[0112]** Para imágenes TFD, el número de bits requerido para señalar el LSB de la LTRP viene determinado por la imagen con el POC más pequeño que preceda, en orden de decodificación, a la imagen actual, y que siga, en orden de decodificación, a la imagen RAP anterior en orden de decodificación, incluyendo la imagen de RAP anterior en orden de decodificación. La estructura sintáctica se basa en Wang et al., "Sobre el conjunto de imágenes de referencia", JCTVC-I0342, contribución a JCTVC, novena reunión: Ginebra, Suiza, del 27 de abril al 7 de mayo de 2012, que a partir del 5 de febrero de 2013 está disponible en [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/9\\_Geneva/wg11/JCTVC-I0342-v1.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/9_Geneva/wg11/JCTVC-I0342-v1.zip), que se incorpora aquí por referencia, donde se señala la longitud de los LSB de la LTRP, seguido por los propios bits LSB. Los cambios en la semántica y en el proceso de decodificación del conjunto de imágenes de referencia, con respecto al documento WD7 de la HEVC, se subrayan en la descripción de este tercer ejemplo de implementación a continuación. Además, debería entenderse que las técnicas de la tercera implementación ejemplar pueden combinarse con técnicas de las implementaciones ejemplares primera y / o segunda, en cualquier combinación.

**[0113]** La Tabla 8 proporciona un conjunto ejemplar de sintaxis para una cabecera de fragmento, de acuerdo a esta tercera implementación ejemplar.

**TABLA 8**

slice_header() {	Descriptor
...	
si (long_term_ref_pics_present_flag) {	
<b>num_long_term_pics</b>	ue(v)
para( i = 0; i < num_long_term_pics; i++ ) {	
<u><b>poc_lsb_len_delta [i]</b></u>	<u>ue(v)</u>
<u><b>poc_lst_lt [i]</b></u>	<u>u(v)</u>
<b>used_by_curr_pic_lt_flag[i]</b>	u(l)
}	
}	
...	
}	

**[0114]** En el ejemplo de la Tabla 8, la cabecera de fragmento incluye los elementos sintácticos adicionales poc\_lsb\_len\_delta [i] y poc\_lst\_lt [i]. La semántica de estos elementos sintácticos se describe a continuación, y la semántica de ciertos elementos sintácticos que se modifican en relación con el documento WD7 también se describen a continuación. La semántica para otros elementos sintácticos puede seguir siendo la misma que en el documento WD7, a menos que se indique lo contrario.

**[0115]** En el ejemplo de la Tabla 8, poc\_lsb\_len\_delta [i] se puede usar para especificar el número de bits para representar poc\_lst\_lt [i].

**[0116]** Sea CurrPicOrderCntVal el PicOrderCntVal de una imagen actual. Las variables PrevLtRefPicPoc y PrevLtRefPicPocForTFD se obtienen de la siguiente manera.

```

si (RapPicFlag || ((CurrPicOrderCntVal) < PrevLtRefPicPoc && (tipo de unidad de NAL != 2 )))
    PrevLtRefPicPoc = CurrPicOrderCntVal
si (RapPicFlag || (CumPicOrderCntVal < PrevLtRefPicPoc))
    PrevLtRefPicPocForTFD = CurrPicOrderCntVal
    
```

**[0117]** La variable MaxPocLsbLstLen puede obtenerse como se indica a continuación.

```

si (nal_unit_type == 2)
    MaxPocLsbLstLen = Log2 (Ceil (CurrPicOrderCntVal - PrevLtRefPicPocForTFD))
en caso contrario
    
```

MaxPocLsbLTLen = Log2 (Ceil (CurrPicOrderCntVal - PrevLtRefPicPoc))

**[0118]** La variable PocLsbLTLen [i] se puede fijar igual a MaxPocLsbLTLen.

5 **[0119]** Alternativamente, la variable PocLsbLTLen [i] se puede obtener de la siguiente manera.

```

si (i = 0)
    PocLsbLTLen [i] = log2_max_pic_order_cnt_lsb_mmus4 + 4 + poc_lsb_len_delta [i]
en caso contrario
10     PocLsbLTLen [i] = PocLsbLTLen [i - 1] + poc_lsb_len_delta [i]
    
```

**[0120]** El valor de PocLsbLTLen [i] puede estar en el rango entre log2\_max\_pic\_order\_cnt\_lsb\_mmus4 + 4 y MaxPocLsbLTLen.

15 **[0121]** Como alternativa, el elemento sintáctico no necesita ser señalado, y la variable PocLsbLTLen [i] puede fijarse igual a MaxPocLsbLTLen.

**[0122]** En el ejemplo de la Tabla 8, poc\_lst\_lt [i] puede especificar los bits menos significativos del valor de recuento de orden de imagen de la i-ésima imagen de referencia a largo plazo que se incluye en el conjunto de imágenes de referencia a largo plazo de la imagen actual. poc\_lst\_lt [i] estará en el intervalo entre 0 y (1 << PocLsbLTLen [i]) - 1, inclusive. La longitud de poc\_lst\_lt [i] es PocLsbLTLen [i].

**[0123]** El decodificador de vídeo 30 puede realizar el siguiente proceso de decodificación para un conjunto de imágenes de referencia de acuerdo a esta tercera implementación ejemplar. El codificador de vídeo 20 puede configurarse para realizar un proceso esencialmente similar mientras codifica datos de vídeo para generar el conjunto de imágenes de referencia. Este proceso puede invocarse una vez por imagen, después de la decodificación de una cabecera de fragmento, pero antes de la decodificación de cualquier unidad de codificación y antes del proceso de decodificación para la construcción de la lista de imágenes de referencia del fragmento, como se especifica en la subcláusula 8.3.3 del documento WD7. El proceso puede dar como resultado marcar una o más imágenes de referencia como "no usada como referencia".

NOTA 1 - El conjunto de imágenes de referencia puede ser una descripción absoluta de las imágenes de referencia utilizadas en el proceso de decodificación de las imágenes codificadas actuales y futuras. La señalización del conjunto de imágenes de referencia puede ser explícita, en el sentido de que todas las imágenes de referencia incluidas en el conjunto de imágenes de referencia se enumeran explícitamente y no existe ningún proceso predeterminado de construcción del conjunto de imágenes de referencia en el decodificador que dependa del estado de la memoria intermedia de imágenes decodificadas.

**[0124]** Las imágenes de referencia a corto plazo pueden identificarse por sus valores PicOrderCntVal. Las imágenes de referencia a largo plazo se pueden identificar por los bits menos significativos de sus valores de PicOrderCntVal.

**[0125]** Se pueden construir cinco listas de valores de recuento de orden de imágenes o de los bits menos significativos de los valores de recuento de orden de imágenes para obtener el conjunto de imágenes de referencia: PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr y PocLtFoll, con los números de elementos NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr y NumPocLtFoll, respectivamente.

50 **[0125]** Si la imagen actual es una imagen de CRA que es la primera imagen codificada en el flujo de bits, una imagen de IDR, o una imagen de BLA, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr y PocLtFoll pueden configurarse como vacíos, y NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr y NumPocLtFoll pueden fijarse en 0.

**[0125]** De lo contrario, puede valer lo siguiente para la obtención de las cinco listas de valores de recuento de orden de imágenes y del número de entradas.

```

55     para (i = 0, j = 0, k = 0; i < NumNegativePics [StRpsIdx]; i++)
        si (UsedByCurrPicS0 [StRpsIdx] [i])
            PocStCurrBefore [j++] = PicOrderCntVal + DeltaPocS0 [StRpsIdx] [i]
        en caso contrario
            PocStFoll [k++] = PicOrderCntVal + DeltaPocS0 [StRpsIdx] [i]
60     NumPocStCurrBefore = j
        para (i = 0, j = 0; i < NumPositivePics [StRpsIdx]; i++)
            si (UsedByCurrPicS1 [StRpsIdx] [i])
                PocStCurrAfter [j++] = PicOrderCntVal + DeltaPocS1 [StRpsIdx] [i]
            en caso contrario
65             PocStFoll [k++] = PicOrderCntVal + DeltaPocS1 [StRpsIdx] [i]
            NumPocStCurrAfter = j
    
```



```

    NumPocStFoll = k (8-5)
    para (i = 0, j = 0, k = 0; i < num_long_term_pics; i ++){
        si (used_by_curr_pic_lt_flag [i] {
            PocLtCurr [i] = poc_lst_lt [i]
            PocLsbLenCurr [i] = PocLsbLtLen [i]
            j++
        }
        en caso contrario {
            PocLtFoll [k] = poc_lsb_lt [i]
            PocLsbLenFoll [k] = PocLsbLtLen [i]
            k++
        }
    }
    NumPocLtCurr = j
    NumPocLtFoll = k

```

donde PicOrderCntVal es el recuento de orden de imagen de la imagen actual como se especifica en la subcláusula 8.2.1 del documento WD7.

NOTA 2 - Un valor de StRpsIdx en el rango de 0 a num\_short\_term\_ref\_pic\_sets - 1, inclusive, puede indicar que se está utilizando un conjunto de imágenes de referencia a corto plazo del conjunto de parámetros de secuencia activa, donde StRpsIdx es el índice del conjunto de imágenes de referencia a corto plazo para la lista de conjuntos de imágenes de referencia a corto plazo en el orden en que se señalizan en el conjunto de parámetros de secuencia. StRpsIdx igual a num\_short\_term\_ref\_pic\_sets puede indicar que se está utilizando un conjunto de imágenes de referencia a corto plazo que se señala explícitamente en la cabecera de fragmento.

**[0126]** El conjunto de imágenes de referencia puede incluir cinco listas de imágenes de referencia: RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetStFoll, RefPicSetLtCurr y RefPicSetLtFoll. La variable NumPocTotalCurr se fija igual a NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr. Al decodificar un fragmento P o B, el documento WD7 requiere, para la conformidad del flujo de bits, que el valor de NumPocTotalCurr no sea igual a 0.

NOTA 3 - El RefPicSetStCurrBefore, el RefPicSetStCurrAfter y el RefPicSetLtCurr pueden contener todas las imágenes de referencia que pueden usarse en la inter-predicción de la imagen actual y que pueden usarse en la inter-predicción de una o más de las imágenes que siguen a la imagen actual en orden de decodificación. El RefPicSetStFoll y el RefPicSetLtFoll pueden incluir todas las imágenes de referencia que no se utilizan en la inter-predicción de la imagen actual, pero se pueden usar en la inter-predicción de una o más de las imágenes siguientes a la imagen actual en orden de decodificación.

**[0127]** El marcado de una imagen de referencia puede ser "no usada para referencia", "usada para referencia a corto plazo" o "usada para referencia a largo plazo", pero solo uno de estos tres en el documento WD7. Cuando se hace referencia a una imagen de referencia marcada como "utilizada como referencia", esto se refiere colectivamente a la imagen marcada como "usada para referencia a corto plazo" o "utilizada para referencia a largo plazo" (pero no ambas en el documento WD7). Una imagen de referencia que está marcada como "utilizada para referencia a corto plazo" se denomina imagen de referencia a corto plazo. Una imagen de referencia que está marcada como "utilizada para referencia a largo plazo" se denomina imagen de referencia a largo plazo.

**[0128]** El proceso de obtención para el conjunto de imágenes de referencia y el marcado de imágenes se llevan a cabo de acuerdo a las siguientes etapas ordenadas, donde DPB se refiere a la memoria intermedia de imágenes decodificadas, como se describe en el Anexo C del documento WD7:

1. Vale lo siguiente:

```

    para(i = 0; i < NumPocLtCurr; i++){
        si (hay una imagen de referencia a largo plazo imagenX en la DPB
            con PicOrderCntVal para el cual GetLSB (PicOrderCntVal, PocLsbLenCurr
            [i]) es igual a PocLtCurr [i])
            RefPicSetLtCurr [i] = imagenX
        en caso contrario, si (hay una imagen de referencia a corto plazo imagenY en la DPB
            con PicOrderCntVal para el cual GetLSB (PicOrderCntVal, PocLsbLenCurr
            [i]) es igual a PocLtCurr [i])
            RefPicSetLtCurr [i] = imagenY
        en caso contrario
            RefPicSetLtCurr [i] = "sin imagen de referencia"
    } (8-6)
    para( i = 0; i < NumPocLtFoll; i++ ) {
        si (hay una imagen de referencia a largo plazo imagenX en la DPB

```

```

    con PicOrderCntVal para el cual GetLSB (PicOrderCntVal, PocLsbLenFoll
    [i]) es igual a PocLtFoll [i])
    RefPicSetLtFoll [i] = imagenX
    en caso contrario, si (hay una imagen de referencia a corto plazo imagenY en la DPB
5     con PicOrderCntVal para el cual GetLSB (PicOrderCntVal, PocLsbLenFoll
    [i]) es igual a PocLtFoll [i])
    RefPicSetLtFoll [i] = imagenY
    en caso contrario
    RefPicSetLtFoll [i] = "sin imagen de referencia"
10  }

```

2. Todas las imágenes de referencia incluidas en el RefPicSetLtCurr y el RefPicSetLtFoll están marcadas como "utilizada para referencia a largo plazo"

15 3. Vale lo siguiente:

```

    para(i = 0; i < NumPocStCurrBefore; i++)
    si (hay una imagen de referencia a corto plazo imagenX en la DPB
    con PicOrderCntVal igual a PocStCurrBefore [i])
20     RefPicSetStCurrBefore [i] = imagenX
    en caso contrario
    RefPicSetStCurrBefore [i] = "sin imagen de referencia"
    para(i = 0; i < NumPocStCurrAfter; i++)
    si (hay una imagen de referencia a corto plazo imagenX en la DPB
25     con PicOrderCntVal igual a PocStCurrAfter [i])
    RefPicSetStCurrAfter [i] = imagenX
    en caso contrario
    RefPicSetStCurrAfter [i] = "sin imagen de referencia" (8-7)
    para(i = 0; i < NumPocStFoll; i++)
30     si (hay una imagen de referencia a corto plazo imagenX en la DPB
    con PicOrderCntVal igual a PocStFoll [i])
    RefPicSetStFoll [i] = imagenX
    en caso contrario
    RefPicSetStFoll [i] = "sin imagen de referencia"
35

```

4. Todas las imágenes de referencia incluidas en RefPicSetStCurrBefore, RefPicSetStCurrAfter y RefPicSetStFoll están marcadas como "utilizada para referencia a corto plazo".

40 5. Todas las imágenes de referencia en la memoria intermedia de imágenes decodificadas que no están incluidas en RefPicSetLtCurr, RefPicSetLtFoll, RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetStFoll están marcadas como "no utilizada como referencia".

45 NOTA 4 - Puede haber una o más imágenes de referencia que están incluidas en el conjunto de imágenes de referencia pero que no están presentes en la memoria intermedia de imágenes decodificadas. Las entradas en RefPicSetStFoll o RefPicSetLtFoll que sean iguales a "sin imagen de referencia" deben ignorarse de acuerdo al documento WD7. A menos que cualquiera de las dos condiciones siguientes sea verdadera, se puede deducir una pérdida de imagen involuntaria para cada entrada en RefPicSetStCurrBefore, RefPicSetStCurrAfter y RefPicSetLtCurr que sea igual a "sin imagen de referencia": a) la primera imagen codificada en el flujo de bits es una imagen de CRA y la imagen codificada actual es una imagen TFD asociada a la primera imagen codificada en el flujo de bits; b) la imagen de RAP anterior que precede a la imagen codificada actual en orden de decodificación es una imagen de BLA y la imagen codificada actual es una imagen TFD asociada a la imagen de BLA.

55 **[0129]** El documento WD7 requiere, para la conformidad del flujo de bits, que el conjunto de imágenes de referencia esté restringido de la siguiente manera, según lo revisado por esta tercera implementación ejemplar:

- 60 - No habrá ninguna imagen de referencia con Id\_temporal mayor que el de la imagen actual incluida en RefPicSetStCurrBefore, RefPicSetStCurrAfter y RefPicSetLtCurr.
- 60 - Cuando la imagen actual es una imagen de TLA, no debe haber ninguna imagen de referencia incluida en el conjunto de imágenes de referencia con Id\_temporal mayor o igual que el Id\_temporal de la imagen actual.
- 65 - No se incluirá ninguna imagen de referencia en el conjunto de imágenes de referencia que preceda, en orden de salida, a cualquier imagen de RAP que preceda a la imagen actual, tanto en orden de decodificación como en orden de salida.

- A menos que cualquiera de las siguientes condiciones sea verdadera, no habrá ninguna entrada en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetLtCurr que sea igual a "sin imagen de referencia": a) la primera imagen codificada en el flujo de bits es una imagen de CRA y la imagen codificada actual es una imagen TFD asociada a la primera imagen codificada en el flujo de bits; b) la imagen de RAP anterior que precede a la imagen codificada actual en el orden de decodificación es una imagen de BLA y la imagen codificada actual es una imagen TFD asociada a la imagen de BLA.

NOTA 5 - Una imagen de referencia no se puede incluir en más de una de las cinco listas de conjuntos de imágenes de referencia.

- Para cada valor de i en el rango de 0 a NumPocLtCurr - 1, inclusive, no habrá más de una imagen de referencia en la DPB que no esté incluida en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetStFoll, y que tenga un PicOrderCntVal para el que GetLSB (PicOrderCntVal) , PocLsbLenCurr [i]) sea igual a PocLtCurr [i]. Para cada valor de i en el rango de 0 a NumPocLtFoll-1, inclusive, no debe haber más de una imagen de referencia en la DPB que no esté incluida en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetStFoll, y que tenga un PicOrderCntVal para el cual GetLSB (PicOrderCntVal, PocLsbLenFoll [i]) sea igual a PocLtFoll [i].
- Sea prevRapPic la imagen de RAP anterior en orden de decodificación. Cuando la imagen actual es una imagen de RAP o precede, en orden de salida, a prevRapPic, no habrá ninguna imagen de referencia incluida en el RefPicSetLtCurr o el RefPicSetLtFoll que preceda, en orden de decodificación, a prevRapPic.

**[0130]** Una cuarta implementación ejemplar proporciona un procedimiento de señalización de LTRP basado en la estructura sintáctica en la tercera implementación ejemplar expuesta anteriormente, donde se señala la longitud de los LSB de la LTRP, seguida de los propios bits LSB. La restricción en las imágenes principales que no hacen referencia a las imágenes antes del RAP asociado se elimina en esta cuarta implementación ejemplar. Las imágenes TFD asociadas a una imagen de RAP están desautorizadas para seguir, en orden de decodificación, a cualquier imagen DLP asociada a la misma imagen de RAP en esta cuarta implementación ejemplar. Para imágenes que no son TFD, el número de bits necesarios para señalar los LSB de la LTRP puede ser determinado por la imagen con el POC más pequeño que no esté marcada como una imagen TFD, y que preceda a la imagen actual, en orden de decodificación, y que siga, en orden de decodificación, a la imagen de RAP anterior en orden de decodificación, incluida la imagen de RAP anterior en orden de decodificación. Para las imágenes TFD, el número de bits necesarios para señalar los LSB de la LTRP puede ser determinado por la imagen con el POC más pequeño que preceda, en orden de decodificación, a la imagen actual y que siga, en orden de salida, al RAP antes de la imagen de RAP anterior en orden de decodificación, incluida la imagen de RAP antes de la imagen de RAP anterior en orden de decodificación. Solo los cambios en la semántica y el proceso de obtención del RPS de la tercera implementación ejemplar se presentan a continuación. Debería entenderse que las técnicas de la cuarta implementación ejemplar pueden combinarse con cualquiera de las técnicas de las implementaciones ejemplares primera, segunda y / o tercera, en cualquier combinación. El texto subrayado representa cambios relativos a la tercera implementación ejemplar descrita anteriormente.

**[0131]** En esta cuarta implementación ejemplar, poc\_lsb\_len\_delta [i] se puede usar para especificar el número de bits para representar poc\_lsb\_len\_delta [i].

**[0132]** Sea CurrPicOrderCntVal el PicOrderCntVal de una imagen actual. Sea prevRapPicPoc el PicOrderCntVal de la imagen de RAP que precede, en orden de decodificación, a la imagen actual. Las variables PrevPrevLtRefPicPoc y PrevLtRefPicPoc se obtienen de la siguiente manera.

```

si (RapPicFlag)
    PrevPrevLtRefPicPoc = prevRapPicPoc
si (RapPicFlag || ((CurrPicOrderCntVal < PrevLtRefPicPoc) && (tipo de unidad de NAL! = 2)))
    PrevLtRefPicPoc = CurrPicOrderCntVal
    
```

**[0133]** La variable MaxPocLsbLtLen puede obtenerse como se indica a continuación.

```

si ((tipo de unidad de NAL = 4) || (tipo de unidad de NAL = 5) || (tipo de unidad de NAL = 2))
    MaxPocLsbLtLen = Log2 (Ceil (CurrPicOrderCntVal - PrevPrevPrevLtRefPicPoc))
en caso contrario
    MaxPocLsbLtLen = Log2 (Ceil (CurrPicOrderCntVal - PrevLtRefPicPoc))
    
```

**[0134]** El proceso de obtención para PocLsbLtLen [i] se puede realizar de la misma manera que en la tercera implementación ejemplar, como se ha descrito anteriormente.

**[0135]** El decodificador de vídeo 30 puede realizar el siguiente proceso de decodificación para un conjunto de imágenes de referencia. El codificador de vídeo 20 puede configurarse para realizar un proceso esencialmente similar mientras codifica datos de vídeo para generar el conjunto de imágenes de referencia. Este proceso puede invocarse una vez por imagen, después de la decodificación de una cabecera de fragmento, pero antes de la

decodificación de cualquier unidad de codificación y antes del proceso de decodificación para la construcción de la lista de imágenes de referencia del fragmento, como se especifica en la subcláusula 8.3.3 del documento WD7. El proceso puede dar como resultado marcar una o más imágenes de referencia como "no usada como referencia".

5       NOTA 1 - El conjunto de imágenes de referencia puede ser una descripción absoluta de las imágenes de referencia utilizadas en el proceso de decodificación de las imágenes codificadas actuales y futuras. La señalización del conjunto de imágenes de referencia es explícita en el sentido de que todas las imágenes de referencia incluidas en el conjunto de imágenes de referencia se enumeran explícitamente y que no existe ningún proceso predeterminado de construcción del conjunto de imágenes de referencia en el decodificador que dependa del estado de la memoria intermedia de imágenes decodificadas.

[0136] En esta cuarta implementación ejemplar, las imágenes de referencia se identifican por sus valores de PicOrderCntVal.

15       [0137] Pueden ser construidas cinco listas de valores de recuento de orden de imagen para obtener el conjunto de imágenes de referencia: PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, y PocLtFoll, con los números de elementos NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr y NumPocLtFoll, respectivamente.

20       [0138] Los puntos suspensivos a continuación representan que el texto puede permanecer igual que en la tercera implementación ejemplar y / o en la versión actual del documento WD7:

- 25       - Si la imagen actual es una imagen de CRA que es la primera imagen codificada en el flujo de bits, una imagen de IDR o una imagen de BLA, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr y PocLtFoll se configuran como vacíos, y NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr y NumPocLtFoll se fijan todos en 0.
- 30       - Para cada valor de i en el rango de 0 a NumPocLtCurr - 1, inclusive, no habrá más de una imagen de referencia en la DPB que no esté incluida en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetStFoll, y que tenga un PicOrderCntVal para el que GetLSB (PicOrderCntVal) , PocLsbLenCurr [i]) sea igual a PocLtCurr [i]. Para cada valor de i en el rango de 0 a NumPocLtFoll - 1, inclusive, no debe haber más de una imagen de referencia en la DPB que no esté incluida en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetStFoll, y que tenga un PicOrderCntVal para el que GetLSB (PicOrderCntVal, PocLsbLenFoll [i]) sea igual a PocLtFoll [i].
- 35       - Cualquier imagen TFD asociada a la imagen de CRA o de BLA deberá preceder a cualquier imagen que no esté marcada como TFD, precede, en orden de salida, a la imagen de CRA o de BLA y sigue, en orden de decodificación, a la imagen de CRA o de BLA.

40       [0139] Una quinta implementación ejemplar también se basa generalmente en la tercera implementación ejemplar. Esta quinta implementación ejemplar divulga un procedimiento que no permite que las imágenes principales asociadas a una imagen de CRA o de BLA se refieran a la imagen de CRA o de BLA asociada, o a otras imágenes principales asociadas a la misma imagen de CRA o de BLA que las LTRP. La restricción, en la tercera implementación ejemplar, en las imágenes principales que no hacen referencia a las imágenes antes del RAP asociado como las LTRP se elimina en esta quinta implementación ejemplar. El número de bits necesarios para señalar los LSB de la LTRP puede ser determinado por la imagen de RAP que precede, en orden de salida, a la imagen actual en esta quinta implementación ejemplar. Además, debería entenderse que las técnicas de esta quinta implementación ejemplar pueden combinarse con cualquiera de las técnicas de las implementaciones ejemplares primera, segunda, tercera y / o cuarta, en cualquier combinación.

50       [0140] En esta quinta implementación ejemplar, la semántica de la cabecera de fragmento de la tercera implementación ejemplar se puede cambiar tal como se indica mediante el texto subrayado a continuación:

[0141] Sea CurrPicOrderCntVal el PicOrderCntVal de una imagen actual. Sea prevRapPicPoc el PicOrderCntVal de la imagen de RAP que precede, en orden de salida, a la imagen actual.

55       [0142] La variable MaxPocLsbLtLen puede obtenerse como se indica a continuación.

$$\text{MaxPocLsbLtLen} = \text{Log2} (\text{Ceil} (\text{CurrPicOrderCntVal}) - \text{prevRapPicPoc})$$

60       [0143] El proceso de obtención para PocLsbLtLen [i] se puede realizar de la misma manera que en la tercera implementación ejemplar.

65       [0144] El decodificador de vídeo 30 puede realizar el siguiente proceso de decodificación para un conjunto de imágenes de referencia. El codificador de vídeo 20 puede configurarse para realizar un proceso esencialmente similar mientras codifica datos de vídeo para generar el conjunto de imágenes de referencia. Este proceso puede invocarse una vez por imagen, después de la decodificación de una cabecera de fragmento, pero antes de la

decodificación de cualquier unidad de codificación y antes del proceso de decodificación para la construcción de la lista de imágenes de referencia del fragmento, como se especifica en la subcláusula 8.3.3 del documento WD7. El proceso puede dar como resultado marcar una o más imágenes de referencia como "no usada como referencia".

5       NOTA 1 - El conjunto de imágenes de referencia puede ser una descripción absoluta de las imágenes de referencia utilizadas en el proceso de decodificación de las imágenes codificadas actuales y futuras. La señalización del conjunto de imágenes de referencia es explícita en el sentido de que todas las imágenes de referencia incluidas en el conjunto de imágenes de referencia se enumeran explícitamente y que no existe ningún proceso predeterminado de construcción del conjunto de imágenes de referencia en el decodificador que dependa del estado de la memoria intermedia de imágenes decodificadas.

[0145] En esta quinta implementación ejemplar, las imágenes de referencia se identifican por sus valores de PicOrderCntVal.

15       [0146] Pueden ser construidas cinco listas de valores de recuento de orden de imagen para obtener el conjunto de imágenes de referencia: PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr, y PocLtFoll, con los números de elementos NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr y NumPocLtFoll, respectivamente.

20       [0147] Los puntos suspensivos a continuación representan que el texto puede permanecer igual que en la tercera implementación ejemplar y / o en la versión actual del documento WD7:

- 25       - Si la imagen actual es una imagen de CRA que es la primera imagen codificada en el flujo de bits, una imagen de IDR o una imagen de BLA, PocStCurrBefore, PocStCurrAfter, PocStFoll, PocLtCurr y PocLtFoll se configuran como vacíos, y NumPocStCurrBefore, NumPocStCurrAfter, NumPocStFoll, NumPocLtCurr y NumPocLtFoll están todos fijados en 0.
- 30       - Para cada valor de i en el rango de 0 a NumPocLtCurr - 1, inclusive, no habrá más de una imagen de referencia en la DPB que no esté incluida en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetStFoll, y que tenga un PicOrderCntVal para el que GetLSB (PicOrderCntVal), PocLsbLenCurr [i]) sea igual a PocLtCurr [i]. Para cada valor de i en el rango de 0 a NumPocLtFoll - 1, inclusive, no debe haber más de una imagen de referencia en el DPB que no esté incluida en RefPicSetStCurrBefore, RefPicSetStCurrAfter o RefPicSetStFoll, y que tenga un PicOrderCntVal para el que GetLSB (PicOrderCntVal, PocLsbLenFoll [i]) sea igual a PocLtFoll [i].
- 35       - Sea prevRapPic la imagen de RAP anterior en orden de decodificación. Cuando la imagen actual precede, en orden de salida, a prevRapPic, no habrá ninguna imagen de referencia incluida en RefPicSetLtCurr o RefPicSetLtFoll, que sea prevRapPic, o cualquier imagen que preceda, en orden de salida, a prevRapPic, pero que siga a prevRapPic, en orden de decodificación.

40       [0148] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse, cada uno, como cualquiera entre una amplia variedad de circuitos codificadores o decodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables en el terreno (FPGA), circuitos de lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cada uno entre el codificador de vídeo 20 y el decodificador de vídeo 30 puede estar incluido en uno o más codificadores o decodificadores, cada uno de los cuales puede estar integrado como parte de un codificador/decodificador (CÓDEC) de vídeo combinado. Un dispositivo que incluye el codificador de vídeo 20 y/o el decodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

50       [0149] La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar técnicas para señalar imágenes de referencia a largo plazo en cabeceras de fragmento. El codificador de vídeo 20 puede realizar la intra-codificación y la inter-codificación de bloques de vídeo dentro de fragmentos de vídeo. La intra-codificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La inter-codificación 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. La intra-modalidad (modalidad I) puede referirse a cualquiera de varias modalidades de codificación de base espacial. Las inter-modalidades, tales como la predicción unidireccional (modalidad P) o la bi-predicción (modalidad B), pueden referirse a cualquiera de varias modalidades de codificación de base temporal.

60       [0150] Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama de vídeo a codificar. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de selección de modalidad 40, una memoria de imágenes de referencia 64, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantización 54 y una unidad de codificación por entropía 56. A su vez, la unidad de selección de modalidad 40 incluye una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una unidad de intra-predicción 46 y una unidad de división 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también una unidad de cuantización inversa 58, una unidad de

procesamiento de transformación inversa 60 y un sumador 62. También puede incluirse un filtro de desbloqueo (no se muestra en la FIG. 2) para filtrar límites de bloque, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de desbloqueo filtrará habitualmente la salida del sumador 62. También pueden usarse filtros adicionales (en bucle o pos-bucle), además del filtro de desbloqueo. Dichos filtros no se muestran por razones de brevedad pero, si se desea, pueden filtrar la salida del sumador 50 (tal como un filtro en bucle).

**[0151]** Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o un fragmento de vídeo que va a codificarse. La trama o el fragmento pueden estar divididos en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 llevan a cabo la codificación inter-predictiva del bloque de vídeo recibido, con respecto a uno o más bloques de una o más tramas de referencia, para proporcionar predicción temporal. La unidad de intra-predicción 46, como alternativa, puede llevar a cabo la codificación intra-predictiva del bloque de vídeo recibido, con respecto a uno o más bloques contiguos de la misma trama o fragmento que el bloque que va a codificarse, para proporcionar predicción espacial. El codificador de vídeo 20 puede llevar a cabo múltiples pasadas de codificación, por ejemplo, para seleccionar una modalidad de codificación adecuada para cada bloque de datos de vídeo.

**[0152]** Además, la unidad de división 48 puede dividir bloques de datos de vídeo en sub-bloques, basándose en la evaluación de los anteriores esquemas de división en las pasadas de codificación anteriores. Por ejemplo, la unidad de división 48 puede dividir inicialmente una trama o un fragmento en varias LCU, y dividir cada una de las LCU en varias sub-CU, basándose en un análisis de velocidad-distorsión (por ejemplo, optimización de velocidad-distorsión). La unidad de selección de modalidad 40 puede producir además una estructura de datos de árbol cuádruple que indica la división de una LCU en varias sub-CU. Las CU de nodos de hojas del árbol cuádruple pueden incluir una o más PU y una o más TU.

**[0153]** La unidad de selección de modalidad 40 puede seleccionar una de las modalidades de codificación, intra o inter, por ejemplo, en función de los resultados de errores, y proporciona el bloque intra-codificado o inter-codificado 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 modalidad 40 proporciona además elementos sintácticos, tales como vectores de movimiento, indicadores de intra-modalidad, información de división y otra información sintáctica de este tipo, a la unidad de codificación por entropía 56.

**[0154]** La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento de 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 o imagen actual de vídeo, con respecto a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada), con respecto al bloque actual que está siendo codificado dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque que se revela como estrechamente coincidente con el bloque a codificar en lo que respecta a la diferencia de píxeles, que puede determinarse mediante la suma de una diferencia absoluta (SAD), una suma de diferencia de cuadrados (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones fraccionarias de píxeles 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 píxel fraccionarias de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxeles completas y las posiciones de píxeles fraccionarias, y emitir un vector de movimiento con una precisión de píxel fraccionaria.

**[0155]** La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento inter-codificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse entre una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44.

**[0156]** La compensación de movimiento, llevada a cabo por la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo basándose en el 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. Tras 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 restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que está siendo codificado, generando valores de diferencia de píxel, como se expone posteriormente. En general, la unidad de estimación de movimiento 42 lleva a cabo la estimación de movimiento con respecto a los componentes de luma, y la unidad de compensación de movimiento 44 utiliza los vectores de movimiento calculados basándose

en los componentes de luma, tanto para los componentes de croma como para los componentes de luma. La unidad de selección de modalidad 40 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por parte del decodificador de vídeo 30 a la hora de decodificar los bloques de vídeo del fragmento de vídeo.

5 [0157] La unidad de selección de modalidad 40 puede implementar las restricciones de esta divulgación, cuando se seleccionan imágenes de referencia de la memoria de imágenes de referencia 64. Por ejemplo, la unidad de selección de modalidad 40 puede determinar si una imagen particular sigue a una imagen que se seleccionó como un RAP. Si es así, la unidad de selección de modalidad 40 puede determinar si la imagen es una imagen principal y sigue a una imagen DLP que está asociada al mismo RAP en el orden de salida. Si es así, la unidad de selección de modalidad 40 selecciona las modalidades de codificación para la imagen de manera que la imagen sea tratada como una imagen DLP. Es decir, la unidad de selección de modalidad 40 asegura que la imagen actual no se predice a partir de ninguna imagen que preceda al RAP ni de cualquier imagen TFD asociada, y así, todas las imágenes TFD preceden a todas las imágenes DLP en orden de salida. En otras palabras, una imagen que de otro modo podría haberse codificado como una imagen TFD se codifica como una imagen DLP si la imagen sigue a otra imagen DLP en orden de codificación. De forma similar, la unidad de selección de modalidad 40 puede seleccionar modalidades de codificación para asegurar que todas las imágenes principales precedan en orden de decodificación a todas las imágenes de cola para un RAP.

20 [0158] La unidad de procesamiento de intra-predicción 46 puede intra-predicir un bloque actual, de forma alternativa a la inter-predicción llevada a cabo por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de procesamiento de intra-predicción 46 puede determinar una modalidad de intra-predicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de intra-predicción 46 puede codificar un bloque actual usando varias modalidades de intra-predicción, por ejemplo, durante diferentes pases de codificación, y la unidad de procesamiento de intra-predicción 46 (o la unidad de selección de modalidad 40, en algunos ejemplos) puede seleccionar una modalidad adecuada de intra-predicción para utilizar a partir de las modalidades probadas.

30 [0159] Por ejemplo, la unidad de procesamiento de intra-predicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para las diversas modalidades de intra-predicción probadas, y seleccionar la modalidad de intra-predicción que tenga las mejores características de velocidad-distorsión entre las modalidades probadas. El análisis de velocidad-distorsión determina en general una magnitud de distorsión (o de errores) entre un bloque codificado y un bloque original, no codificado, que se codificó para generar el bloque codificado, así como una velocidad binaria (es decir, un cierto número de bits) usada para generar el bloque codificado. La unidad de procesamiento de intra-predicción 46 puede calcular razones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modalidad de intra-predicción presenta el mejor valor de velocidad-distorsión para el bloque.

40 [0160] Después de seleccionar una modalidad de intra-predicción para un bloque, la unidad de intra-predicción 46 puede proporcionar información, indicativa de la modalidad de intra-predicción seleccionada para el bloque, a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información indicando la modalidad de intra-predicción seleccionada. El codificador de vídeo 20 puede incluir datos de configuración en el flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índices de modalidades de intra-predicción y una pluralidad de tablas de índices de modalidades de intra-predicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones de la modalidad de intra-predicción más probable, una tabla de índices de modalidades de intra-predicción y una tabla de índices de modalidades de intra-predicción modificadas a utilizar para cada uno de los contextos.

50 [0161] El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción, de la unidad de selección de modalidad 40, al bloque de vídeo original que está codificándose. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformaciones 52 aplica una transformación, tal como una transformación discreta del coseno (DCT) o una transformación conceptualmente similar, al bloque residual, generando un bloque de vídeo que comprende valores residuales de coeficientes de transformación. La unidad de procesamiento de transformaciones 52 puede llevar a cabo otras transformaciones que son conceptualmente similares a la DCT. También podrían usarse transformaciones de ondículas, transformaciones de enteros, transformaciones de sub-bandas u otros tipos de transformaciones. En cualquier caso, la unidad de procesamiento de transformaciones 52 aplica la transformación al bloque residual, generando un bloque de coeficientes de transformación residuales. La transformación puede convertir la información residual, desde un dominio de valores de píxel a un dominio de transformaciones, tal como un dominio de frecuencia. La unidad de procesamiento de transformaciones 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantización 54. La unidad de cuantización 54 cuantiza los coeficientes de transformación para reducir adicionalmente la velocidad de bits. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos o a la totalidad de los coeficientes. El grado de cuantización puede modificarse ajustando un parámetro de cuantización. En algunos ejemplos, la unidad de cuantización 54 puede realizar, a continuación, un recorrido de la matriz que incluye los coeficientes de transformación cuantizados. De

forma alternativa, la unidad de codificación por entropía 56 puede llevar a cabo el recorrido.

**[0162]** Tras la cuantización, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformación cuantizados. Por ejemplo, la unidad de codificación por entropía 56 puede llevar a cabo la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto y basada en sintaxis (SBAC), la codificación por entropía mediante la división en intervalos de probabilidades (PIPE) u otra técnica de codificación por entropía. En el caso de la codificación por entropía basada en el contexto, el contexto puede basarse en bloques contiguos. Tras la codificación por entropía realizada por la unidad de codificación por entropía 56, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, el decodificador de vídeo 30) o archivarse para su posterior transmisión o recuperación.

**[0163]** La unidad de cuantización inversa 58 y la unidad de procesamiento de transformación inversa 60 aplican la cuantización inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio del píxel, por ejemplo, para su posterior uso como bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo 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 fraccionarios y usarlos en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, generado por la unidad de compensación de movimiento 44, para generar un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede ser usado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para inter-codificar un bloque en una trama de vídeo posterior.

**[0164]** El codificador de vídeo 20 de la FIG. 2 representa un ejemplo de un codificador de vídeo que puede configurarse para realizar cualquiera de las técnicas de esta divulgación. Por ejemplo, el codificador de vídeo 20 puede configurarse para codificar una imagen de RAP y codificar una o más imágenes principales decodificables, DLP, para la imagen de RAP, de manera que todas las imágenes que están destinadas a descartar precedan a las DLP asociadas a la imagen de RAP en orden de visualización. Las DLP pueden incluir una o más imágenes que tienen valores de orden de visualización anteriores a un valor de orden de visualización de la imagen de RAP y que no se refieren a datos de vídeo anteriores a la imagen de RAP en orden de decodificación. El codificador de vídeo 20 también puede codificar una o más imágenes principales con respecto a la imagen de RAP, de manera que todas las imágenes principales para la imagen de RAP precedan a todas las imágenes de cola de la imagen de RAP en orden de decodificación, donde las imágenes de cola incluyen imágenes que tienen valores de orden de visualización que son mayores que un valor de orden de visualización de la imagen de RAP. La imagen de RAP puede incluir, por ejemplo, una entre una imagen de CRA y una imagen de BLA. Cualquier imagen que preceda a una imagen de CRA o de BLA en orden de decodificación puede preceder a cualquier imagen DLP asociada a la imagen de CRA o a la imagen de BLA en orden de visualización. El codificador de vídeo 20 también puede codificar una o más imágenes principales asociadas a la imagen de RAP, donde las imágenes principales preceden a la imagen de RAP en valor de orden de visualización y siguen a la imagen de RAP en orden de decodificación, y el codificador de vídeo 20 puede codificar una o más imágenes de cola asociadas a la imagen de RAP de modo que todas las imágenes principales precedan a todas las imágenes de cola en orden de decodificación, donde las imágenes de cola siguen a la imagen de RAP, tanto en orden de visualización como en orden de decodificación.

**[0165]** El codificador de vídeo 20 también puede codificar una o más imágenes etiquetadas para descartar (TFD), asociadas a la imagen de RAP. Las una o más imágenes TFD pueden preceder a la imagen de RAP en el valor de orden de visualización y seguir a la imagen RAP en orden de decodificación y hacer referencia a los datos de vídeo anteriores a la imagen de RAP en el orden de decodificación. El codificador de vídeo 20 puede codificar las una o más DLP para la imagen de RAP codificando las una o más DLP asociadas a la imagen de RAP de manera que todas las imágenes TFD precedan a las DLP en orden de visualización, donde las DLP preceden a la imagen de RAP en orden de visualización y siguen a la imagen de RAP en el orden de decodificación y no hacen referencia a los datos de vídeo anteriores a la imagen de RAP en el orden de decodificación.

**[0166]** El codificador de vídeo 20 también puede codificar una o más imágenes que preceden a la imagen de RAP en un orden de decodificación. El codificador 20 de vídeo puede codificar una o más DLP codificando las una o más DLP asociadas a la imagen de RAP, de manera que todas las imágenes que preceden a la imagen de RAP en orden de decodificación también precedan a todas las DLP en orden de visualización. Las DLP pueden preceder a la imagen de RAP en orden de visualización y seguir a la imagen de RAP en orden de decodificación y no hacer referencia a los datos de vídeo anteriores a la imagen de RAP en orden de decodificación.

**[0167]** La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 que puede implementar técnicas para señalar imágenes de referencia a largo plazo en cabeceras de fragmento. En el ejemplo de la FIG. 3, el decodificador de vídeo 30 incluye una unidad de decodificación por entropía 70, una unidad de compensación de movimiento 72, una unidad de intra-predicción 74, una unidad de cuantización inversa 76, una unidad de transformación inversa 78, una memoria de imágenes de referencia 82 y un sumador 80. En algunos ejemplos, el decodificador de vídeo 30 puede llevar a cabo una pasada de decodificación generalmente recíproca a



la pasada de codificación descrita con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción basados en vectores de movimiento recibidos desde la unidad de decodificación por entropía 70, mientras que la unidad de intra-predicción 74 puede generar datos de predicción basados en indicadores de modalidad de intra-predicción recibidos desde la unidad de decodificación por entropía 70.

**[0168]** Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado, que representa los bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de decodificación por entropía 70 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantizados, vectores de movimiento o indicadores de modalidad de intra-predicción y otros elementos sintácticos. La unidad de decodificación por entropía 70 remite los vectores de movimiento y otros elementos sintácticos a la unidad de compensación de movimiento 72. El decodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel del fragmento de vídeo y/o el nivel del bloque de vídeo.

**[0169]** Cuando el fragmento de vídeo se codifica como un fragmento intra-codificado (I), la unidad de intra-predicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en una modalidad de intra-predicción señalizada, y datos de bloques previamente decodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento inter-codificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, basándose en los vectores de movimiento y en otros elementos sintácticos recibidos desde la unidad de decodificación por entropía 70. Los bloques predictivos pueden ser generados 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, la Lista 0 y la Lista 1, usando técnicas de construcción por omisión, basándose en las imágenes de referencia almacenadas en la memoria de imágenes de referencia 92. La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos del bloque de vídeo actual que está siendo decodificado. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, intra-predicción o inter-predicción), usada para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de inter-predicción (por ejemplo, fragmento B, fragmento P o fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia del fragmento, vectores de movimiento para cada bloque de vídeo inter-codificado del fragmento, el estado de inter-predicción para cada bloque de vídeo inter-codificado del fragmento y otra información para decodificar los bloques de vídeo en el fragmento de vídeo actual.

**[0170]** La unidad de compensación de movimiento 72 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles fraccionarios de bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación utilizados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos, y utilizar los filtros de interpolación para generar bloques predictivos.

**[0171]** La unidad de cuantización inversa 76 cuantiza de manera inversa, es decir, descuantiza, los coeficientes de transformación cuantizados, proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 80. El proceso de cuantización inversa puede incluir el uso de un parámetro de cuantización  $QP_V$ , calculado por el decodificador de vídeo 30 para cada bloque de vídeo en el fragmento de vídeo para determinar un grado de cuantización y, asimismo, un grado de cuantización inversa que debería aplicarse.

**[0172]** La unidad de procesamiento de transformación inversa 78 aplica una transformación inversa, por ejemplo una DCT inversa, una transformación inversa entera o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación, con el fin de generar bloques residuales en el dominio de píxeles.

**[0173]** Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformación inversa 78 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72. El sumador 90 representa el componente o los componentes que llevan a cabo esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques decodificados con el fin de eliminar distorsiones de efecto pixelado. También pueden utilizarse otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para allanar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo decodificados en una trama o imagen dada son a continuación almacenados en la memoria de imágenes de referencia 92, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La memoria de imágenes de referencia 82 almacena también vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

**[0174]** Cuando se implementan técnicas de esta divulgación, el decodificador 30 de vídeo puede recibir una imagen

de RAP y determinar que la imagen de RAP se usa como un punto de acceso aleatorio para un flujo de bits. La determinación de que la imagen de RAP va a ser utilizada por el decodificador de vídeo 30 como un punto de acceso aleatorio puede, por ejemplo, basarse en la señalización en el flujo de bits. Para las imágenes que siguen a la imagen de RAP en orden de codificación, el decodificador de vídeo puede determinar si las imágenes son imágenes TFD. Si una imagen es una imagen TFD, entonces el decodificador de vídeo 30 puede analizar la sintaxis asociada a la imagen TFD sin necesariamente decodificar la imagen TFD. Por ejemplo, basándose en un tipo de unidad de NAL, el decodificador de vídeo puede identificar imágenes TFD y no decodificar completamente la imagen TFD. Si el decodificador de vídeo 30 determina que una imagen es una DLP, entonces el decodificador de vídeo 30 puede decodificar la imagen.

**[0175]** El decodificador de vídeo 30 de la FIG. 3 representa un ejemplo de un codificador de vídeo que puede configurarse para realizar cualquiera de las técnicas de esta divulgación. Por ejemplo, el codificador de vídeo 20 puede configurarse para codificar una imagen de RAP y codificar una o más imágenes principales decodificables, DLP, para la imagen de RAP, de manera que todas las imágenes que están destinadas a descartar precedan a las DLP asociadas a la imagen de RAP en orden de visualización. Las DLP pueden incluir una o más imágenes que tienen valores de orden de visualización anteriores a un valor de orden de visualización de la imagen de RAP y que no se refieren a datos de vídeo anteriores a la imagen de RAP en orden de decodificación. El decodificador de vídeo 30 también puede codificar una o más imágenes principales con respecto a la imagen de RAP, de manera que todas las imágenes principales de la imagen de RAP precedan a todas las imágenes de cola de la imagen de RAP en orden de decodificación, donde las imágenes de cola incluyen imágenes con valores de orden de visualización que son mayores que un valor de orden de visualización de la imagen de RAP. La imagen de RAP puede incluir, por ejemplo, una entre una imagen de CRA y una imagen de BLA. Cualquier imagen que preceda a una imagen de CRA o de BLA en orden de decodificación puede preceder a cualquier imagen DLP asociada a la imagen de CRA o a la imagen de BLA en orden de visualización. El codificador de vídeo 20 también puede codificar una o más imágenes principales asociadas a la imagen de RAP, donde las imágenes principales preceden a la imagen de RAP en valor de orden de visualización y siguen a la imagen de RAP en orden de decodificación, y el decodificador 30 puede codificar una o más imágenes finales asociadas a la imagen de RAP de modo que todas las imágenes principales precedan a todas las imágenes de cola en orden de decodificación, donde las imágenes de cola siguen a la imagen de RAP, tanto en orden de visualización como en orden de decodificación.

**[0176]** El decodificador de vídeo 30 también puede codificar una o más imágenes etiquetadas para descartar (TFD) asociadas a la imagen de RAP. Las una o más imágenes TFD pueden preceder a la imagen de RAP en el valor de orden de visualización y seguir a la imagen RAP en orden de decodificación y hacer referencia a los datos de vídeo anteriores a la imagen de RAP en el orden de decodificación. El codificador de vídeo 20 puede codificar las una o más DLP para la imagen de RAP codificando las una o más DLP asociadas a la imagen de RAP de manera que todas las imágenes TFD precedan a las DLP en orden de visualización, donde las DLP preceden a la imagen de RAP en orden de visualización y siguen a la imagen de RAP en el orden de decodificación y no hacen referencia a los datos de vídeo anteriores a la imagen de RAP en el orden de decodificación.

**[0177]** El decodificador de vídeo 30 también puede codificar una o más imágenes que preceden a la imagen de RAP en orden de decodificación. El decodificador de vídeo 30 puede codificar una o más DLP codificando las una o más DLP asociadas a la imagen de RAP de manera que todas las imágenes que preceden a la imagen de RAP en orden de decodificación también precedan a todas las DLP en orden de visualización. Las DLP pueden preceder a la imagen de RAP en orden de visualización y seguir a la imagen de RAP en orden de decodificación y no hacer referencia a los datos de vídeo anteriores a la imagen de RAP en orden de decodificación.

**[0178]** La FIG. 4 es un diagrama conceptual que ilustra una secuencia de imágenes de vídeo codificadas 100-132. La FIG. 4 ilustra las relaciones entre los RAP, las imágenes principales, las TFD y las imágenes de cola. Las imágenes están sombreadas de manera diferente para indicar las posiciones dentro de una estructura de predicción jerárquica. Por ejemplo, las imágenes 100, 116 y 132 están sombreadas en negro para representar que las imágenes 100, 116, 132 están en la parte superior de la estructura de predicción jerárquica. Las imágenes 100, 116, 132 pueden comprender, por ejemplo, imágenes intra-codificadas o imágenes inter-codificadas que se predicen a partir de otras imágenes en una única dirección (por ejemplo, imágenes P). Cuando están intra-codificadas, las imágenes 100, 116, 132 se predicen únicamente a partir de datos dentro de la misma imagen. Cuando se inter-codifican, la imagen 116, por ejemplo, puede codificarse con relación a los datos de la imagen 100, como se indica mediante la flecha discontinua desde la imagen 116 a la imagen 100. Las imágenes 116, 132 forman imágenes clave de los grupos de imágenes (GOP) 134, 136, respectivamente.

**[0179]** Las imágenes 108, 124 están oscurecidas para indicar que están a continuación en la jerarquía de codificación a continuación de las imágenes 100, 116 y 132. Las imágenes 108, 124 pueden comprender imágenes bidireccionales codificadas con modalidad de inter-predicción. Por ejemplo, la imagen 108 puede predecirse a partir de los datos de las imágenes 100 y 116, mientras que la imagen 124 puede predecirse a partir de las imágenes 116 y 132. Las imágenes 104, 112, 120 y 128 están ligeramente sombreadas para indicar que están a continuación en la jerarquía de codificación que sigue a las imágenes 108 y 124. Las imágenes 104, 112, 120 y 128 también pueden comprender imágenes bidireccionales codificadas con modalidad de inter-predicción. Por ejemplo, la imagen 104 puede predecirse a partir de las imágenes 100 y 108, la imagen 112 puede predecirse a partir de las imágenes 108 y

116, la imagen 120 puede predecirse a partir de las imágenes 116 y 124, y la imagen 128 puede predecirse a partir de la imagen 124 y 132. En general, las imágenes que están más abajo en la jerarquía pueden codificarse a partir de cualquier imagen de referencia que sea superior en la jerarquía, suponiendo que las imágenes de referencia todavía están almacenadas temporalmente en una memoria intermedia de imágenes decodificadas, y suponiendo que las imágenes de referencia fueron codificadas antes que la imagen que actualmente está siendo codificada.

**[0180]** Finalmente, las imágenes 102, 106, 110, 114, 118, 122, 126 y 130 son sombreadas de color blanco para indicar que estas imágenes son las últimas en la jerarquía de codificación. Las imágenes 102, 106, 110, 114, 118, 122, 126 y 130 pueden ser imágenes bidireccionales codificadas con modalidad de inter-predicción. La imagen 102 puede predecirse a partir de las imágenes 100 y 104, la imagen 106 puede predecirse a partir de las imágenes 104 y 108, la imagen 110 puede predecirse a partir de las imágenes 108 y 112, la imagen 114 puede predecirse a partir de las imágenes 112 y 116, la imagen 118 puede predecirse a partir de las imágenes 116 y 120, la imagen 122 puede predecirse a partir de las imágenes 120 y 124, la imagen 126 puede predecirse a partir de las imágenes 124 y 128, y la imagen 130 puede predecirse a partir de las imágenes 128 y 132. Nuevamente, debe entenderse que las imágenes inferiores en la jerarquía de codificación pueden codificarse a partir de otras imágenes que están más alto en la jerarquía de codificación. Por ejemplo, cualquiera de, o todas, las imágenes 102, 106, 110 o 114 pueden predecirse en relación con cualquiera de las imágenes 100, 116 o 108, además o como alternativa.

**[0181]** Las imágenes 100-132 se ilustran en orden de visualización. Es decir, después de la decodificación, la imagen 100 se visualiza antes que la imagen 102, la imagen 102 se visualiza antes que la imagen 104, y así sucesivamente. Como se ha expuesto anteriormente, los valores de POC generalmente describen un orden de visualización de imágenes, que también es esencialmente el mismo que el orden en el que las imágenes en bruto fueron capturadas o generadas antes de ser codificadas. Sin embargo, debido a la jerarquía de codificación, las imágenes 100-132 pueden decodificarse en un orden diferente. Además, mientras son codificadas, las imágenes 100-132 pueden disponerse en orden de decodificación en un flujo de bits que incluye datos codificados para las imágenes 100-132. Por ejemplo, la imagen 116 se puede mostrar última entre las imágenes del GOP 134. Sin embargo, debido a la jerarquía de codificación, la imagen 116 puede decodificarse como la primera del GOP 134. Es decir, para decodificar adecuadamente la imagen 108, por ejemplo, la imagen 116 puede necesitar ser decodificada primero, para actuar como una imagen de referencia para la imagen 108. Asimismo, la imagen 108 puede actuar como una imagen de referencia para las imágenes 104, 106, 110 y 112, y por lo tanto puede necesitar ser decodificada antes que las imágenes 104, 106, 110 y 112.

**[0182]** Además, ciertas imágenes pueden tratarse como imágenes de referencia a largo plazo, mientras que otras imágenes pueden tratarse como imágenes de referencia a corto plazo. Supongamos, por ejemplo, que las imágenes 100 y 116 representan imágenes de referencia a largo plazo, mientras que las imágenes 108, 104 y 112 representan imágenes de referencia a corto plazo. Puede ser el caso, en este ejemplo, que las imágenes 102 y 106 puedan predecirse con respecto a cualquiera de las imágenes 100, 116, 108 o 104, pero que las imágenes 110 y 114 puedan predecirse con relación a cualquiera de las imágenes 100, 116, 108 o 112. En otras palabras, la imagen 104 puede no estar disponible para referencia cuando se codifican las imágenes 110 y 114. Como otro ejemplo, suponiendo que las imágenes 100 y 116 representan imágenes de referencia a largo plazo y las imágenes 108, 104 y 112 representan imágenes de referencia a corto plazo, las imágenes 108, 104 y 112 pueden no estar disponibles como referencia cuando se codifican las imágenes 118, 122, 126 y 130. De acuerdo a las técnicas de esta divulgación, los datos con respecto a las imágenes de referencia a largo plazo pueden señalizarse en las cabeceras de fragmento de los fragmentos para cualquiera de, o todas, las imágenes 100-132.

**[0183]** En el ejemplo de la FIG. 4, las flechas representan posibles relaciones de predicción, donde el extremo de cada flecha representa una imagen que puede predecirse, y la cabeza de cada flecha representa una imagen a partir de la cual se puede predecir la imagen (es decir, la punta de la flecha señala una potencial imagen de referencia). Aunque la imagen 116 tiene una flecha que apunta a la imagen 100, para fines de ejemplo, supongamos que la imagen 116 está codificada por inter-predicción (es decir, codificada sin referencia a ninguna otra imagen). Además, supongamos que la imagen 108 está codificada como una imagen P con referencia a la imagen 116. Además, supongamos que la imagen 116 se está utilizando como un punto de acceso aleatorio, por ejemplo, como una imagen de IDR, BLA o CRA. Es decir, un flujo de bits recuperado a partir de la imagen 116 incluiría cada una de las imágenes 102-132 en orden de decodificación, pero no incluiría la imagen 100.

**[0184]** En este ejemplo, bajo los supuestos establecidos anteriormente, las imágenes 102, 104 y 106 representan imágenes TFD, porque las imágenes 102, 104 y 106 dependen, directa o indirectamente, de la imagen 100. Las imágenes 108-114 representan ejemplos de DLP, porque las imágenes 108-114 no dependen directa o indirectamente de la imagen 100, y tienen un orden de visualización anterior a la imagen 116, pero un orden de decodificación posterior a la imagen 116. Las imágenes 118-132 representan imágenes de cola en este ejemplo, porque el orden de visualización y el orden de decodificación de las imágenes 118-132 son posteriores a la imagen 116. De acuerdo a las técnicas de esta divulgación, cuando se codifican una o más DLP para una imagen de RAP, las DLP se pueden codificar de manera que todas las imágenes que están destinadas a descartarse precedan a las DLP asociadas a la imagen de RAP en orden de visualización. Por ejemplo, como se ha descrito anteriormente, con referencia a la FIG. 2, cuando se codifica la imagen 108, la unidad de selección de modalidad 40 puede determinar el código de la imagen 108 como una imagen P relativa, por ejemplo, a la imagen 116. De esta manera, la imagen

108 puede clasificarse como una DLP. En consecuencia, la unidad de selección de modalidad 40 puede determinar codificar cada una de las imágenes 110, 112, 114 como DLP también, en lugar de como TFD. De manera similar, el decodificador de vídeo 30 puede, al determinar que la imagen 108 es una DLP, determinar que las imágenes subsiguientes en el orden de salida son decodificables. Por otro lado, el decodificador de vídeo 30 puede simplemente analizar sintácticamente, sin decodificar, datos para cualquier imagen que sea TFD. En este ejemplo, el decodificador de vídeo 30 puede determinar que las imágenes 102, 104 y 106 son imágenes TFD, y por lo tanto, simplemente analizar sintácticamente los datos de estas imágenes sin intentar decodificar las imágenes. Para las imágenes que tienen orden de salida posterior a la imagen 108, el decodificador de vídeo 30 puede comenzar a decodificar normalmente, sin determinar si las imágenes son TFD o DLP, porque, de acuerdo a las restricciones propuestas en esta divulgación, todas las imágenes TFD preceden a todas las imágenes DLP en orden de salida.

**[0185]** Como se ha expuesto anteriormente, esta descripción propone varias restricciones para dar soporte al acceso aleatorio en base a estos diversos tipos de imágenes. En primer lugar, todas las imágenes TFD para una imagen de RAP deben tener un valor de orden de visualización que sea anterior al valor de orden de visualización para todas las DLP para el RAP. Esta restricción puede mejorar la experiencia de visualización de un espectador, ya que evita el encrespamiento en la velocidad de tramas. Es decir, si hay una o más imágenes que no se pueden decodificar correctamente (es decir, imágenes TFD), si estas imágenes tienen valores de orden de visualización entremezclados con las DLP, la velocidad de tramas aparecerá entrecortada, lo que disminuiría la experiencia del usuario. Esta divulgación también propone una restricción en la que se evita el entrelazado de imágenes principales con "imágenes de cola" en orden de decodificación. Es decir, de acuerdo a las técnicas de esta divulgación, los codificadores de vídeo pueden asegurar que todas las imágenes principales (incluyendo tanto imágenes TFD como las DLP) tengan valores de orden de decodificación anteriores a los valores de orden de decodificación de imágenes que tengan valores de orden de visualización mayores que el valor de orden de visualización de la imagen de RAP correspondiente. Como se ha indicado anteriormente, las "imágenes de cola" son imágenes que siguen a la imagen del RAP, tanto en orden de decodificación como en orden de visualización.

**[0186]** La FIG. 5 es un diagrama de bloques que ilustra un conjunto ejemplar de dispositivos que forman parte de la red 150. En este ejemplo, la red 150 incluye los dispositivos de encaminamiento 154A, 154B (dispositivos de encaminamiento 154) y el dispositivo de transcodificación 156. Los dispositivos de encaminamiento 154 y el dispositivo de transcodificación 156 están concebidos para representar un pequeño número de dispositivos que pueden formar parte de la red 150. Otros dispositivos de red, tales como conmutadores, concentradores, pasarelas, cortafuegos, puentes y otros dispositivos de ese tipo también pueden estar incluidos dentro de la red 150. Además, pueden proporcionarse dispositivos de red adicionales a lo largo de un trayecto de red entre el dispositivo servidor 152 y el dispositivo cliente 158. El dispositivo servidor 152 puede corresponder al dispositivo de origen 12 (FIG. 1), mientras que el dispositivo cliente 158 puede corresponder al dispositivo de destino 14 (FIG. 1), en algunos ejemplos.

**[0187]** En general, los dispositivos de encaminamiento 154 implementan uno o más protocolos de encaminamiento para intercambiar datos de red a través de la red 150. En general, los dispositivos de encaminamiento 154 ejecutan protocolos de encaminamiento para descubrir rutas a través de la red 150. Al ejecutar tales protocolos de encaminamiento, el dispositivo de encaminamiento 154B puede descubrir una ruta de red desde sí mismo hasta el dispositivo servidor 152, mediante el dispositivo de encaminamiento 154A.

**[0188]** La FIG. 6 es un diagrama de flujo que ilustra una técnica ejemplar de esta divulgación. Las técnicas de la FIG. 6 se describirán con referencia a un codificador de vídeo genérico. Este codificador de vídeo genérico puede, por ejemplo, corresponder al codificador de vídeo 20 o al decodificador de vídeo 30. El codificador de vídeo codifica una imagen de RAP (161). La imagen de RAP puede incluir una entre una imagen limpia de acceso aleatorio y una imagen de acceso de enlace roto. El codificador de vídeo codifica una o más DLP para la imagen de RAP, de modo que todas las imágenes que están destinadas a descarte precedan a las DLP asociadas a la imagen de RAP en orden de visualización (162). Las DLP pueden incluir una o más imágenes que tengan valores de orden de visualización anteriores a un valor de orden de visualización de la imagen de RAP y que no se refieran a datos de vídeo anteriores a la imagen de RAP en orden de decodificación.

**[0189]** La FIG. 7 es un diagrama de flujo que ilustra una técnica ejemplar de esta divulgación. Las técnicas de la FIG. 7 se describirán con referencia a un codificador de vídeo genérico. Este codificador de vídeo genérico puede, por ejemplo, corresponder al codificador de vídeo 20 o al decodificador de vídeo 30. El codificador de vídeo codifica una imagen de RAP (171). El codificador de vídeo codifica una o más imágenes TFD asociadas a la imagen de RAP, donde las imágenes TFD preceden a la imagen de RAP en orden de visualización y siguen a la imagen de RAP en decodificación y hacen referencia a datos de vídeo anteriores a la imagen de RAP en orden de decodificación (172). El codificador de vídeo codifica una o más DLP asociadas a la imagen de RAP de modo que todas las imágenes TFD precedan a las DLP en orden de visualización, donde las DLP preceden a la imagen de RAP en orden de visualización y siguen a la imagen de RAP en decodificación y no se refieren a datos de vídeo anteriores a la imagen de RAP en orden de decodificación (173).

**[0190]** La FIG. 8 es un diagrama de flujo que ilustra una técnica ejemplar de esta divulgación. Las técnicas de la FIG. 8 se describirán con referencia a un codificador de vídeo genérico. Este codificador de vídeo genérico puede,

por ejemplo, corresponder al codificador de vídeo 20 o al decodificador de vídeo 30. El codificador de vídeo codifica una imagen de RAP (181). El codificador de vídeo codifica una o más imágenes principales asociadas a la imagen de RAP, donde las imágenes principales preceden a la imagen de RAP en orden de visualización y siguen a la imagen de RAP en orden de decodificación (182). El codificador de vídeo codifica una o más imágenes de cola asociadas a la imagen de RAP de manera que todas las imágenes principales precedan a todas las imágenes de cola en orden de decodificación, en donde las imágenes de cola siguen a la imagen de RAP, tanto en orden de visualización como en orden de decodificación (183).

**[0191]** La FIG. 9 es un diagrama de flujo que ilustra una técnica ejemplar de esta divulgación. Las técnicas de la FIG. 9 se describirán con referencia a un codificador de vídeo genérico. Este codificador de vídeo genérico puede, por ejemplo, corresponder al codificador de vídeo 20 o al decodificador de vídeo 30. El vídeo codifica una o más imágenes que preceden a una imagen de RAP en orden de decodificación (191). El codificador de vídeo codifica la imagen de RAP (192). El codificador de vídeo codifica una o más DLP asociadas a la imagen de RAP de forma tal que todas las imágenes que preceden a la imagen de RAP en orden de decodificación también precedan a todas las DLP en orden de visualización, donde las DLP preceden a la imagen de RAP en orden de visualización y siguen a la imagen de RAP en decodificación y no hacen referencia a datos de vídeo anteriores a la imagen de RAP en orden de decodificación (193).

**[0192]** La FIG. 10 es un diagrama de flujo que ilustra una técnica ejemplar de esta divulgación. Las técnicas de la FIG. 10 pueden, por ejemplo, ser realizadas por la unidad de selección de modalidad 40 del codificador de vídeo 20. La unidad de selección de modalidad 40 puede codificar una imagen de RAP (202). Para una siguiente imagen, después de la imagen de RAP, la unidad de selección de modalidad 40 puede determinar si codificará la siguiente imagen como una TFD o una DLP (204). Si la unidad de selección de modalidad 40 codifica la siguiente imagen como una TFD (206), entonces, para una imagen posterior, la unidad de selección de modalidad 40 puede determinar nuevamente si codifica la imagen como una TFD o una DLP (204). Una vez que la unidad de selección de modalidad 40 codifica una imagen después de una imagen de RAP como una DLP (208), la unidad de selección de modalidad codifica las imágenes siguientes como DLP (210). En el ejemplo de la FIG. 10, "siguiente imagen" generalmente se refiere a una imagen que es posterior en orden de salida. El entrelazado del orden de decodificación de las DLP y TFD todavía está permitido, mientras que el entrelazado del orden de salida puede ser desautorizado.

**[0193]** La FIG. 11 es un diagrama de flujo que ilustra una técnica ejemplar de esta divulgación. Las técnicas de la FIG. 11 pueden, por ejemplo, ser realizadas por el decodificador de vídeo 30. El decodificador de vídeo 30 puede decodificar una imagen de RAP y determinar que la imagen se va a usar para el acceso aleatorio a un flujo de bits. Para la siguiente imagen en el flujo de bits, el decodificador de vídeo 30 puede determinar, por ejemplo, basándose en un tipo de unidad de NAL para la imagen, si la imagen es una imagen TFD o una DLP (214). Si la imagen es una TFD, entonces el decodificador de vídeo 30 puede omitir la decodificación de la imagen TFD (216). Si la imagen es una DLP, entonces el decodificador de vídeo 30 puede decodificar la DLP (218).

**[0194]** Aunque las técnicas de las FIGs. 6, 7, 8, 9, 10 y 11 se han presentado por separado, se contempla que los aspectos de las técnicas presentadas en las FIGs. 6, 7, 8, 9, 10 y 11 pueden implementarse al mismo tiempo. También debe reconocerse que, según el ejemplo, ciertos actos o sucesos de cualquiera de las técnicas descritas en el presente documento pueden realizarse en una secuencia distinta, pueden añadirse, fundirse u omitirse por completo (por ejemplo, no todos los actos o sucesos descritos son necesarios para la puesta en práctica de las técnicas). Además, en ciertos ejemplos, los actos o sucesos pueden realizarse simultáneamente, por ejemplo, mediante el procesamiento de múltiples hebras, el procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

**[0195]** En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones, como una o más instrucciones o código, pueden almacenarse en, y transmitirse por, un medio legible por ordenador, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos o unos medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) unos medios de almacenamiento tangibles legibles por ordenador que son no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

**[0196]** A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda utilizarse para almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que

5 pueda accederse mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, tal como se utiliza en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, donde algunos discos normalmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior deberían incluirse también dentro del alcance de los medios legibles por ordenador.

15 **[0197]** Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones lógicas programables in situ (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

25 **[0198]** Las técnicas de la esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por medio de un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como los descritos anteriormente, conjuntamente con software y/o firmware adecuados.

35 **[0199]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

**REIVINDICACIONES**

1. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:
  - 5 codificar (161) una imagen de punto de acceso aleatorio, RAP; en el que la imagen de RAP comprende una entre una imagen de acceso limpio aleatorio, CRA, y una imagen de acceso de enlace roto, BLA;
 

10 codificar una o más imágenes principales omitidas de acceso aleatorio, RASL, asociadas a la imagen de RAP, en donde las una o más imágenes RASL preceden a la imagen de RAP en el valor de orden de salida y siguen a la imagen de RAP en orden de decodificación y hacen referencia a datos de vídeo anteriores a la imagen de RAP en el orden de decodificación; codificar (162) una o más imágenes principales decodificables de acceso aleatorio, RADL, para la imagen de RAP, de acuerdo a una restricción de que todas las imágenes RASL precedan a las RADL asociadas a la imagen de RAP en orden de salida, donde las RADL comprenden una o más imágenes que tienen valores de orden de salida que indican un orden de salida anterior a un valor de orden de salida de la imagen de RAP y que tienen valores de orden de decodificación que indican un orden de decodificación posterior a un valor de orden de decodificación de la imagen de RAP, y en el que las una o más imágenes no hacen referencia a datos de vídeo anteriores a la imagen de RAP en orden de decodificación.
  - 20 2. El procedimiento de la reivindicación 1, que comprende además codificar una o más imágenes principales con respecto a la imagen de RAP, de acuerdo a una restricción de que todas las imágenes principales para la imagen de RAP precedan a todas las imágenes de cola para la imagen de RAP en orden de decodificación, en donde las imágenes de cola comprenden imágenes que tienen valores de orden de salida que son mayores que un valor de orden de salida de la imagen de RAP.
  - 25 3. El procedimiento de la reivindicación 1, en el que cualquier imagen que preceda a una imagen de CRA o BLA en orden de decodificación precede a cualquier imagen RADL asociada a la imagen de CRA o a la imagen de BLA en orden de salida.
  - 30 4. El procedimiento de la reivindicación 1, comprendiendo además el procedimiento:
    - 35 codificar una o más imágenes principales asociadas a la imagen de RAP, en donde las imágenes principales preceden a la imagen de RAP en el valor de orden de salida y siguen a la imagen de RAP en orden de decodificación; y
 

40 codificar una o más imágenes de cola asociadas a la imagen de RAP de acuerdo a una restricción de que todas las imágenes principales precedan a todas las imágenes de cola en orden de decodificación, en donde las imágenes de cola siguen a la imagen de RAP, tanto en valor de orden de salida como en orden de decodificación.
  - 45 5. El procedimiento de la reivindicación 1, en el que codificar una o más RADL para la imagen de RAP comprende codificar una o más RADL asociadas a la imagen de RAP de acuerdo a la restricción de que todas las imágenes RASL precedan a las RADL en orden de salida, donde las RADL preceden a la imagen de RAP en orden de salida y siguen a la imagen de RAP en orden de decodificación y no hacen referencia a datos de vídeo anteriores a la imagen de RAP en el orden de decodificación.
  - 50 6. El procedimiento de la reivindicación 1, comprendiendo además el procedimiento:
    - 55 codificar una o más imágenes que preceden a la imagen de RAP en orden de decodificación; y
 

donde codificar las una o más RADL comprende codificar las una o más RADL asociadas a la imagen de RAP de acuerdo a una restricción de que todas las imágenes que preceden a la imagen de RAP en orden de decodificación también precedan a todas las RADL en orden de salida, donde las RADL preceden a la imagen de RAP en orden de salida y siguen a la imagen de RAP en orden de decodificación, y no hacen referencia a los datos de vídeo anteriores a la imagen de RAP en orden de decodificación.
  7. El procedimiento de la reivindicación 1, en el que la codificación de la imagen de RAP se realiza como parte de un proceso de codificación de vídeo.
  - 60 8. Un procedimiento para decodificar datos de vídeo codificados de acuerdo al procedimiento de una cualquiera de las reivindicaciones 1 a 7, comprendiendo el procedimiento:
    - 65 decodificar una imagen de punto de acceso aleatorio, RAP;
 

decodificar una o más imágenes principales omitidas de acceso aleatorio, RASL, asociadas a la imagen de RAP; y

decodificar una o más imágenes principales decodificables de acceso aleatorio, RADL, para la imagen de RAP de acuerdo a una restricción de que todas las una o más imágenes RASL precedan a las RADL asociadas a la imagen de RAP en orden de salida.

- 5
9. Un aparato para codificar datos de vídeo (20), comprendiendo el aparato:
- medios para codificar una imagen de punto de acceso aleatorio, RAP;
- 10 en el que la imagen de RAP comprende una entre una imagen de acceso aleatorio limpio, CRA, y una imagen de acceso de enlace roto, BLA;
- medios para codificar una o más imágenes principales omitidas de acceso aleatorio, RASL, asociadas a la imagen de RAP, donde las una o más imágenes RASL preceden a la imagen de RAP en el valor de orden de salida y siguen a la imagen de RAP en orden de decodificación y hacen referencia a datos de vídeo anteriores a la imagen de RAP en el orden de decodificación; y,
- 15
- medios para codificar una o más imágenes principales decodificables de acceso aleatorio, RADL, para la imagen de RAP de acuerdo a una restricción de que todas las imágenes RASL precedan a las RADL asociadas a la imagen de RAP en orden de salida,
- 20
- donde las RADL comprenden una o más imágenes que tienen valores de orden de salida que indican un orden de salida anterior a un valor de orden de salida de la imagen de RAP y que tienen valores de orden de decodificación que indican un orden de decodificación posterior a un valor de orden de decodificación de la imagen de RAP, y donde las una o más imágenes no hacen referencia a datos de vídeo anteriores a la imagen de RAP en orden de decodificación.
- 25
10. El aparato de la reivindicación 9, que comprende además medios para codificar una o más imágenes principales con respecto a la imagen de RAP, de acuerdo a una restricción de que todas las imágenes principales para la imagen de RAP precedan a todas las imágenes de cola para la imagen de RAP en orden de decodificación, en donde las imágenes de cola comprenden imágenes que tienen valores de orden de salida que son mayores que un valor de orden de salida de la imagen de RAP.
- 30
11. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando son ejecutadas por uno o más procesadores, hacen que los uno o más procesadores lleven a cabo el procedimiento de acuerdo a una cualquiera de las reivindicaciones 1 a 8.
- 35



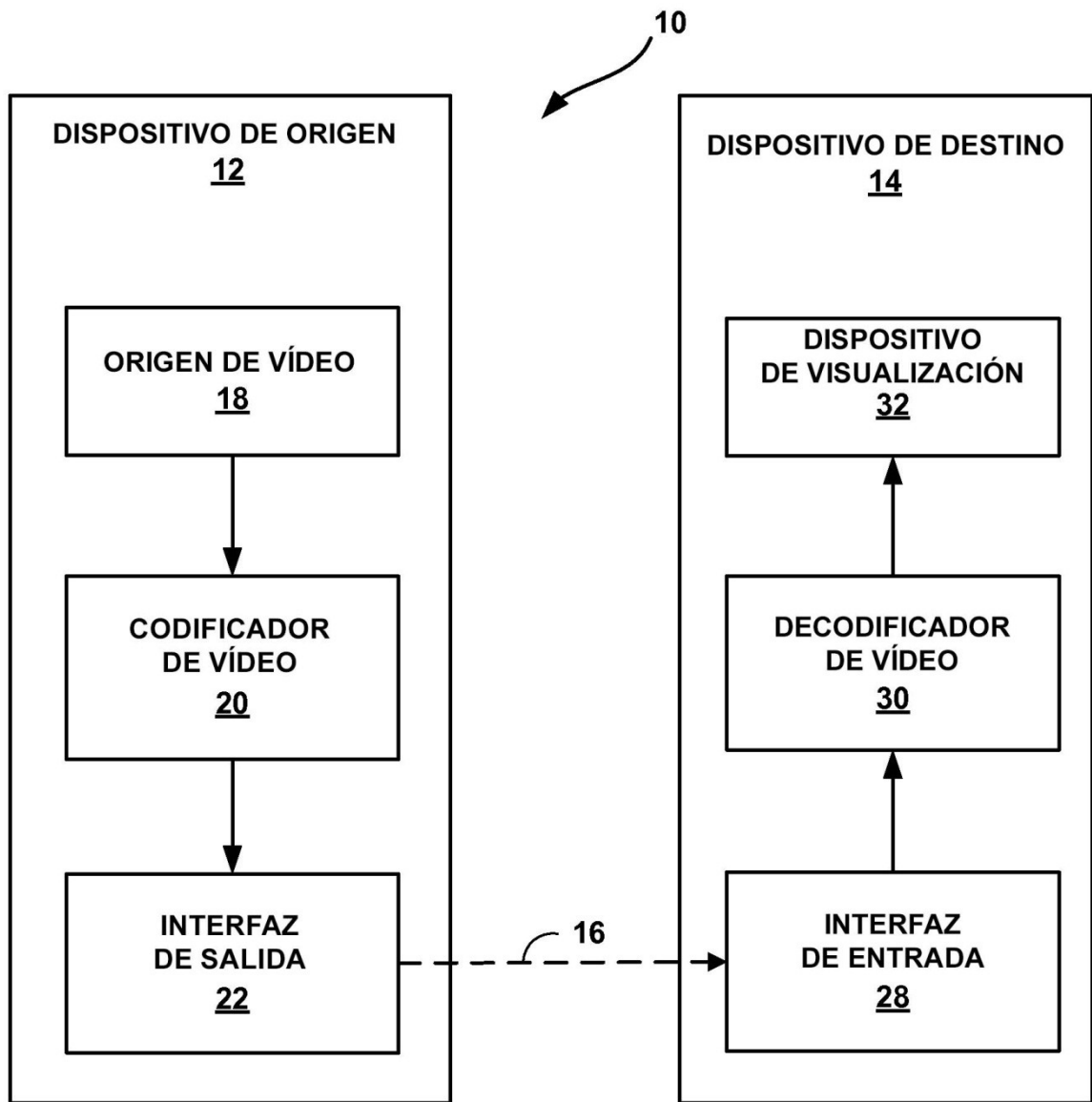


FIG. 1

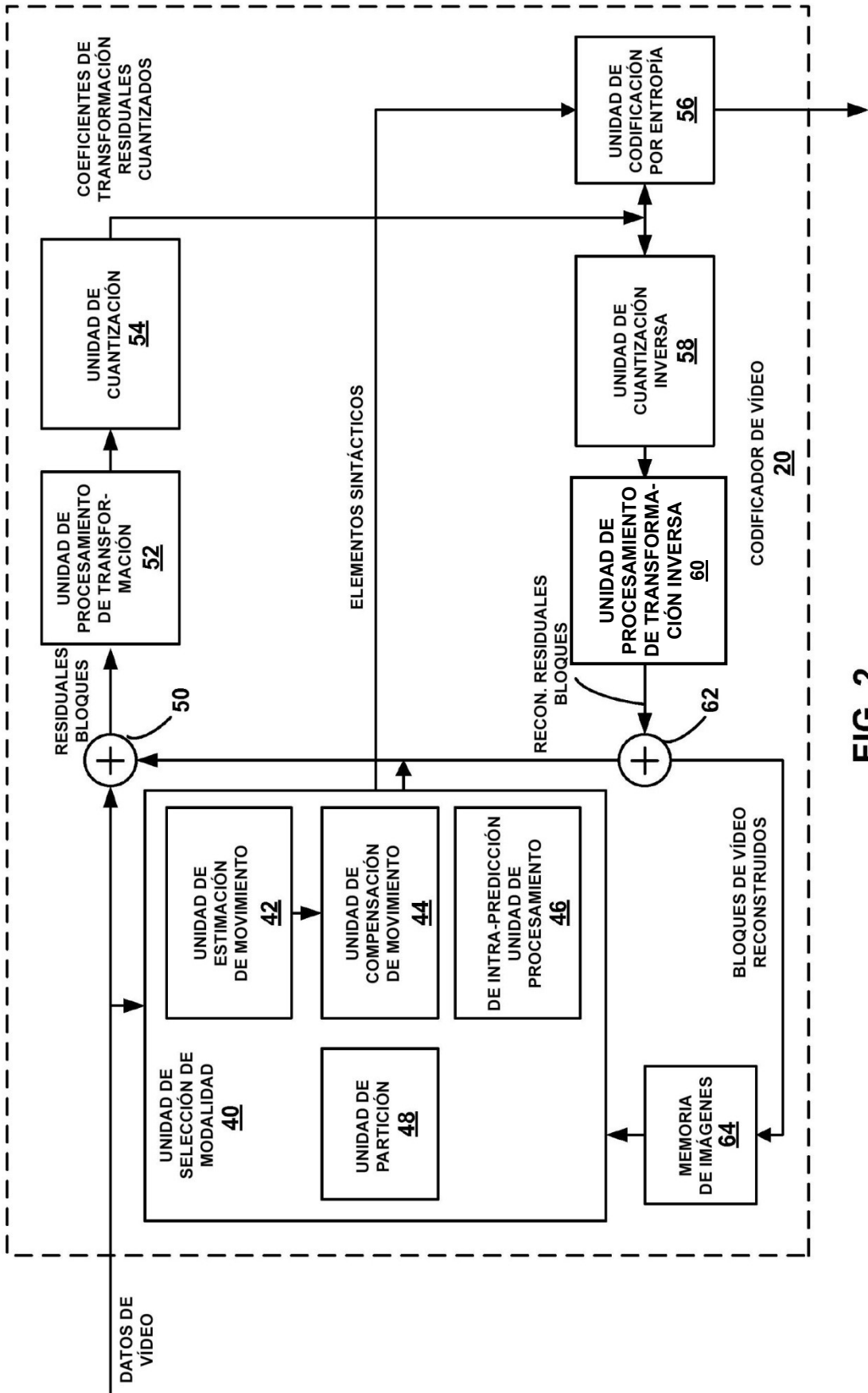


FIG. 2

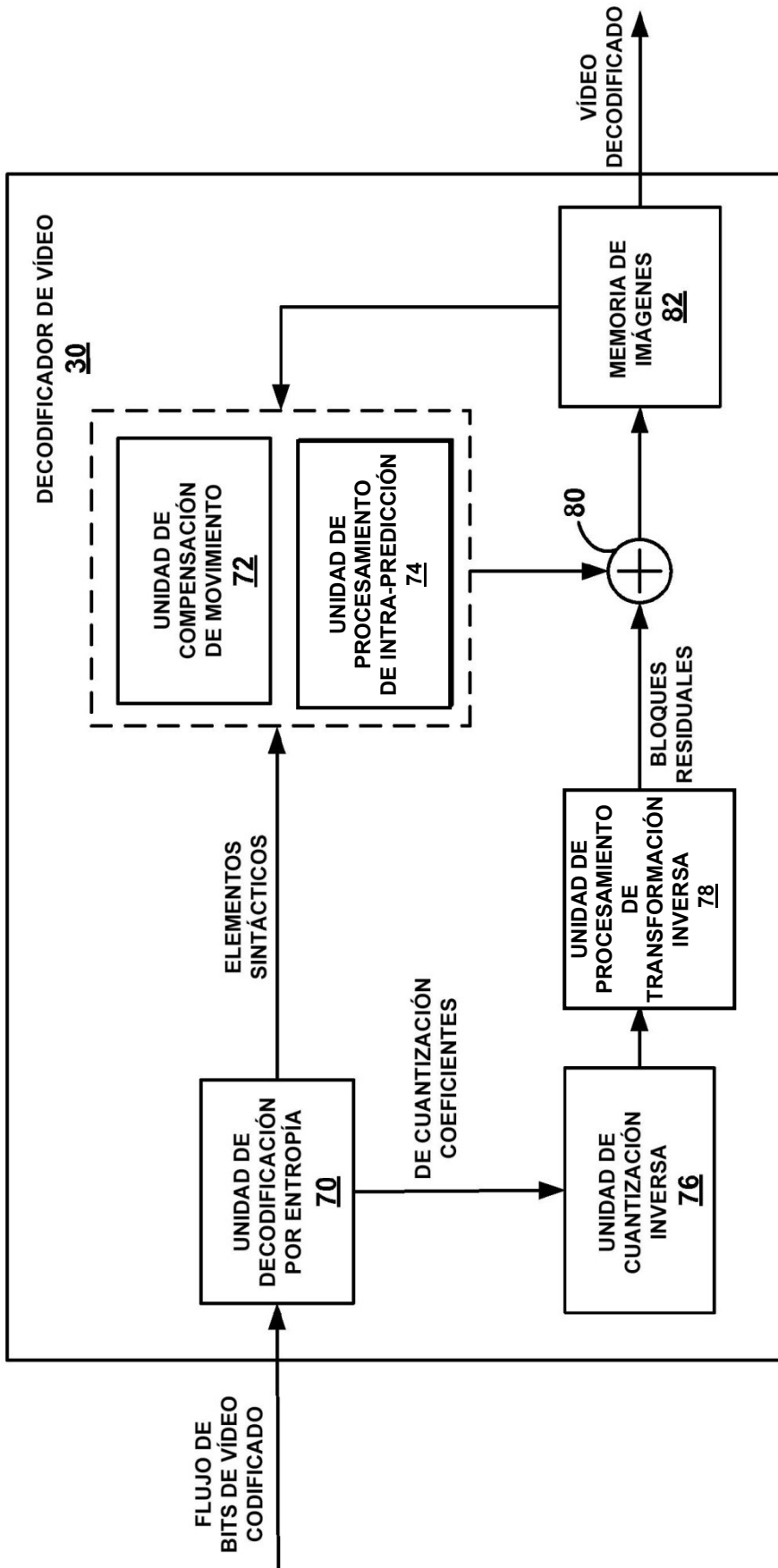


FIG. 3

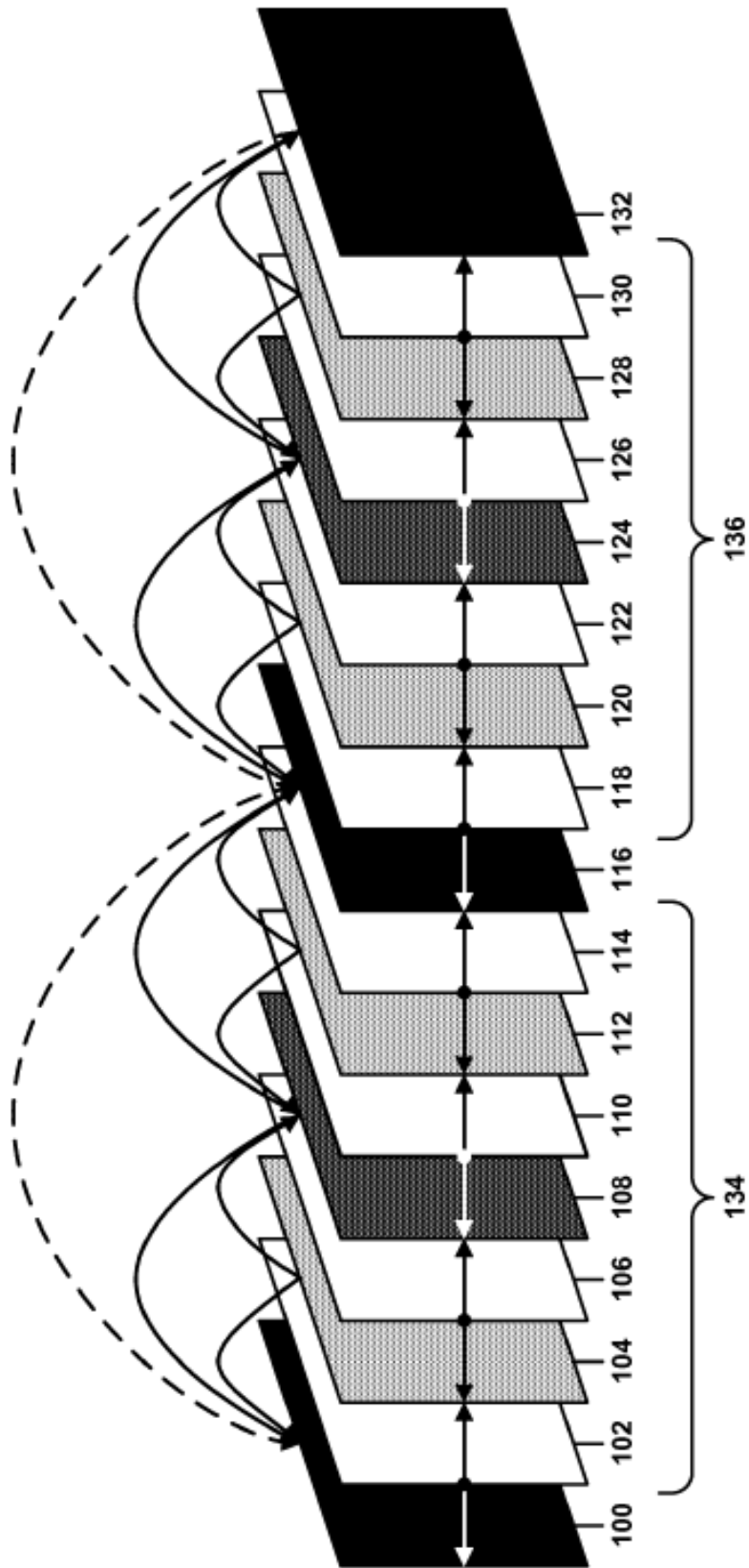


FIG. 4

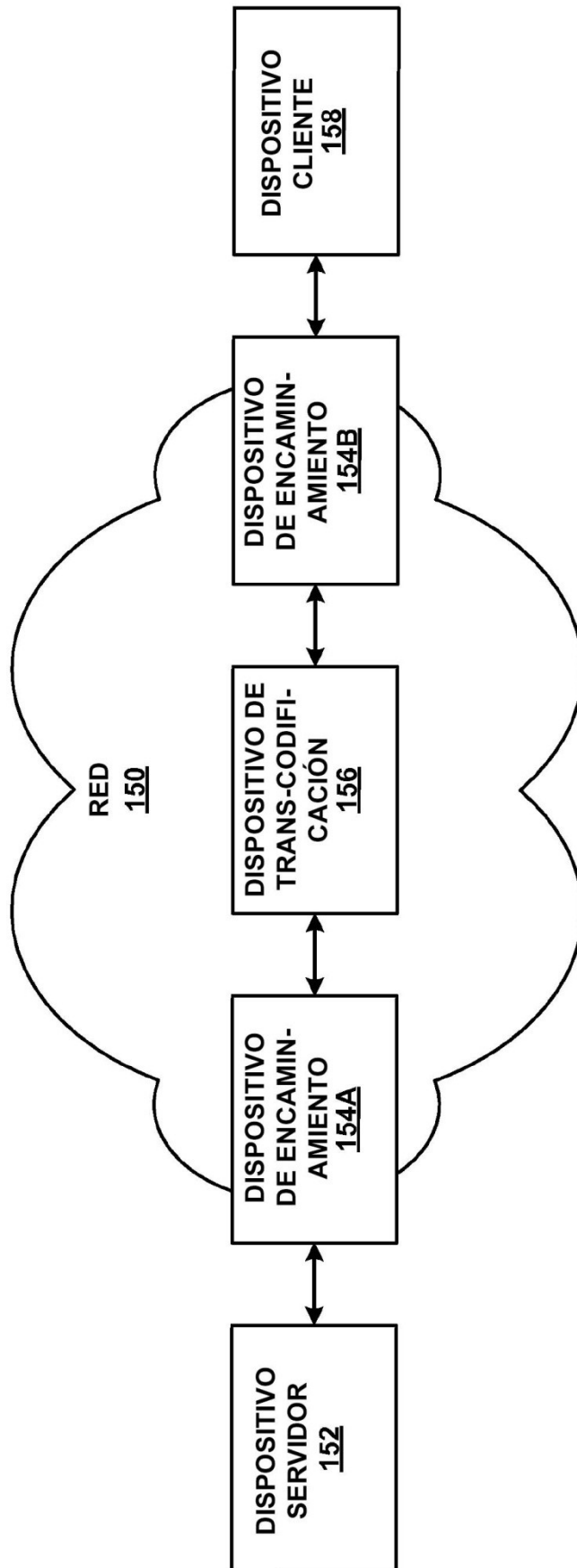
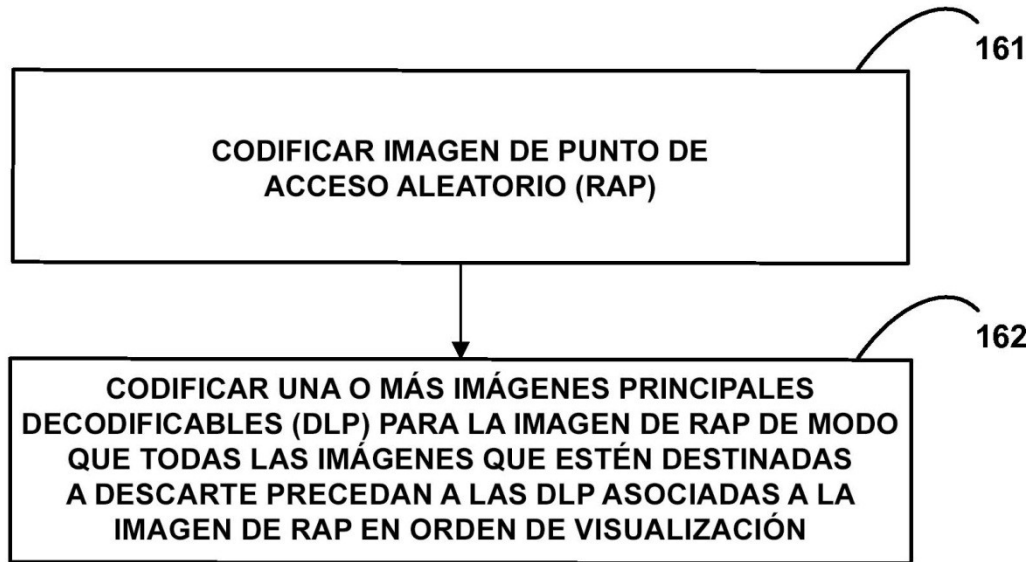
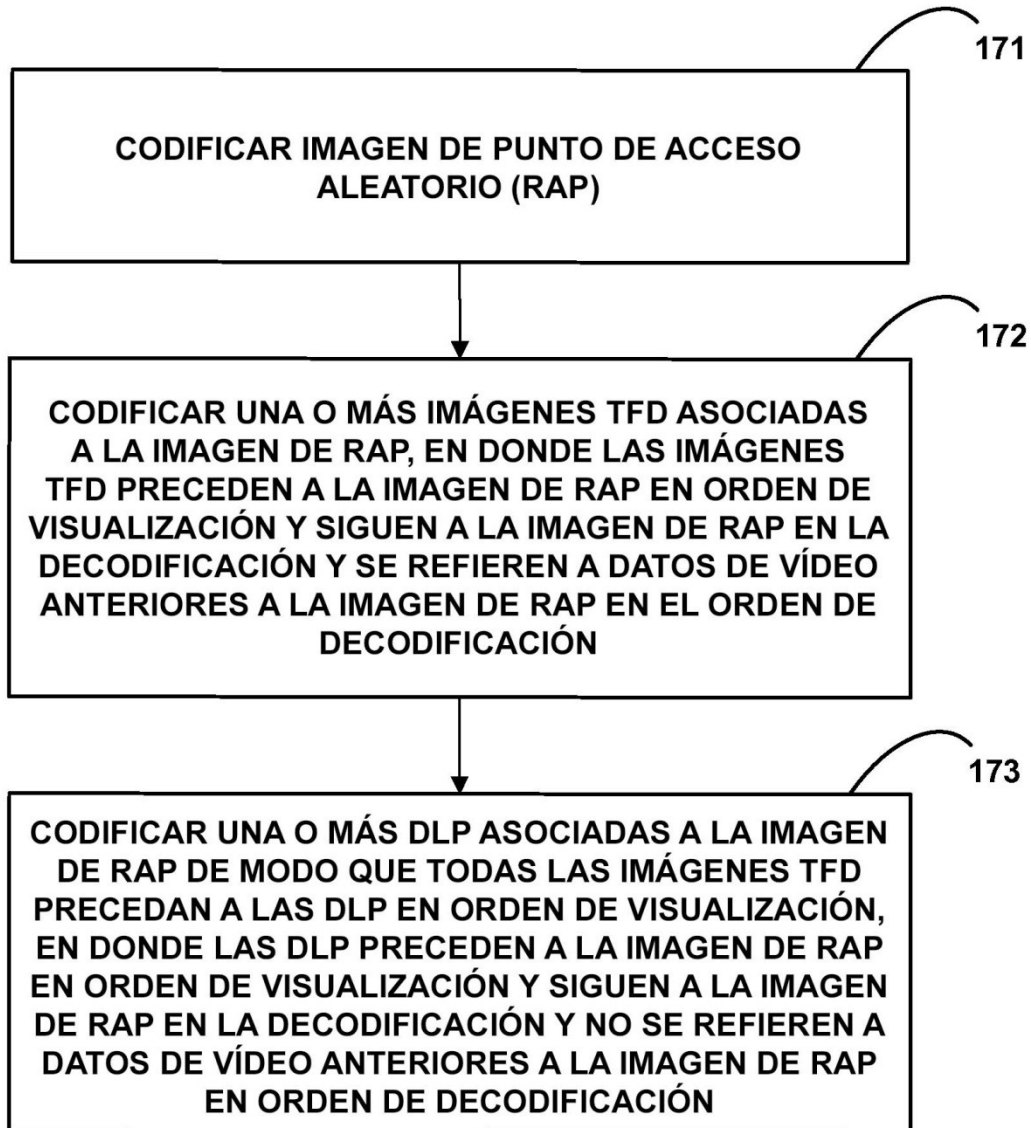


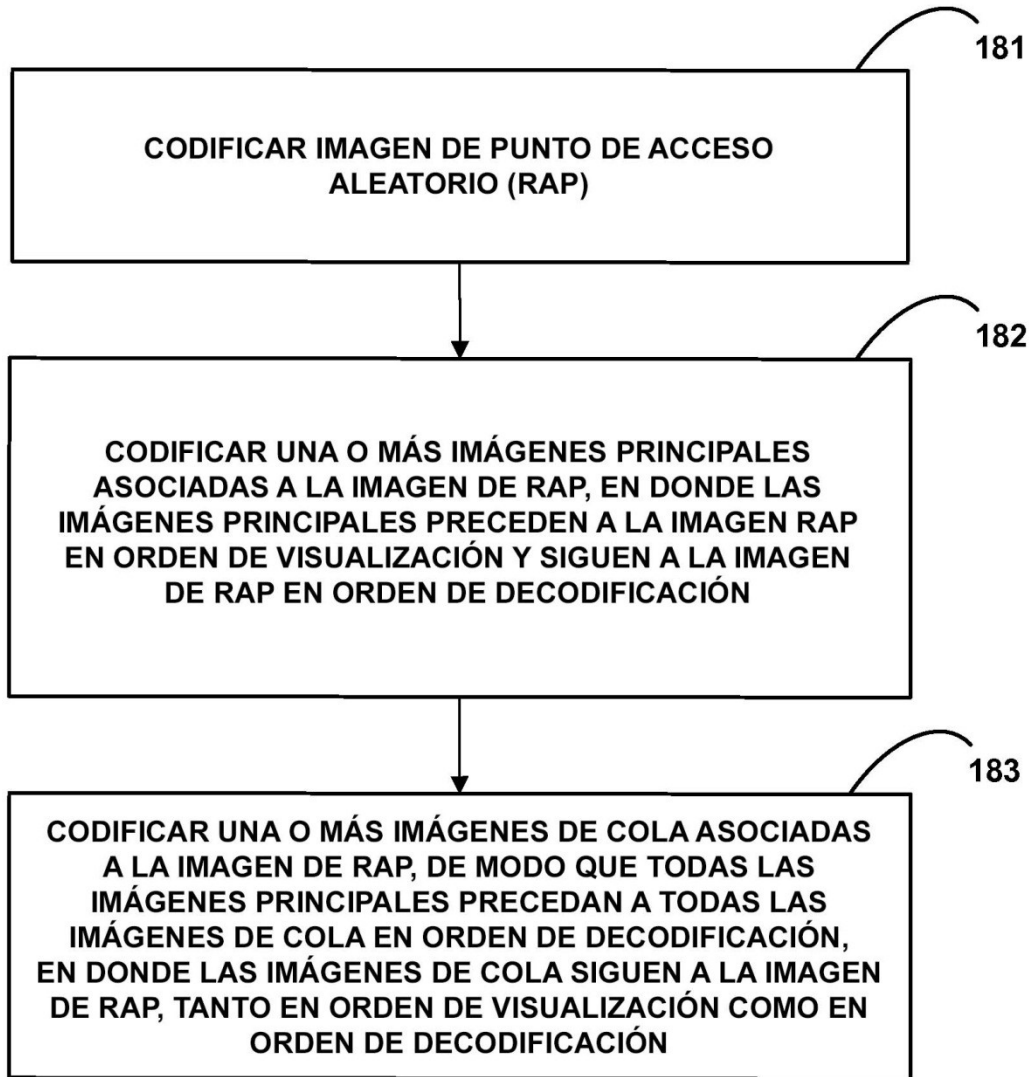
FIG. 5



**FIG. 6**

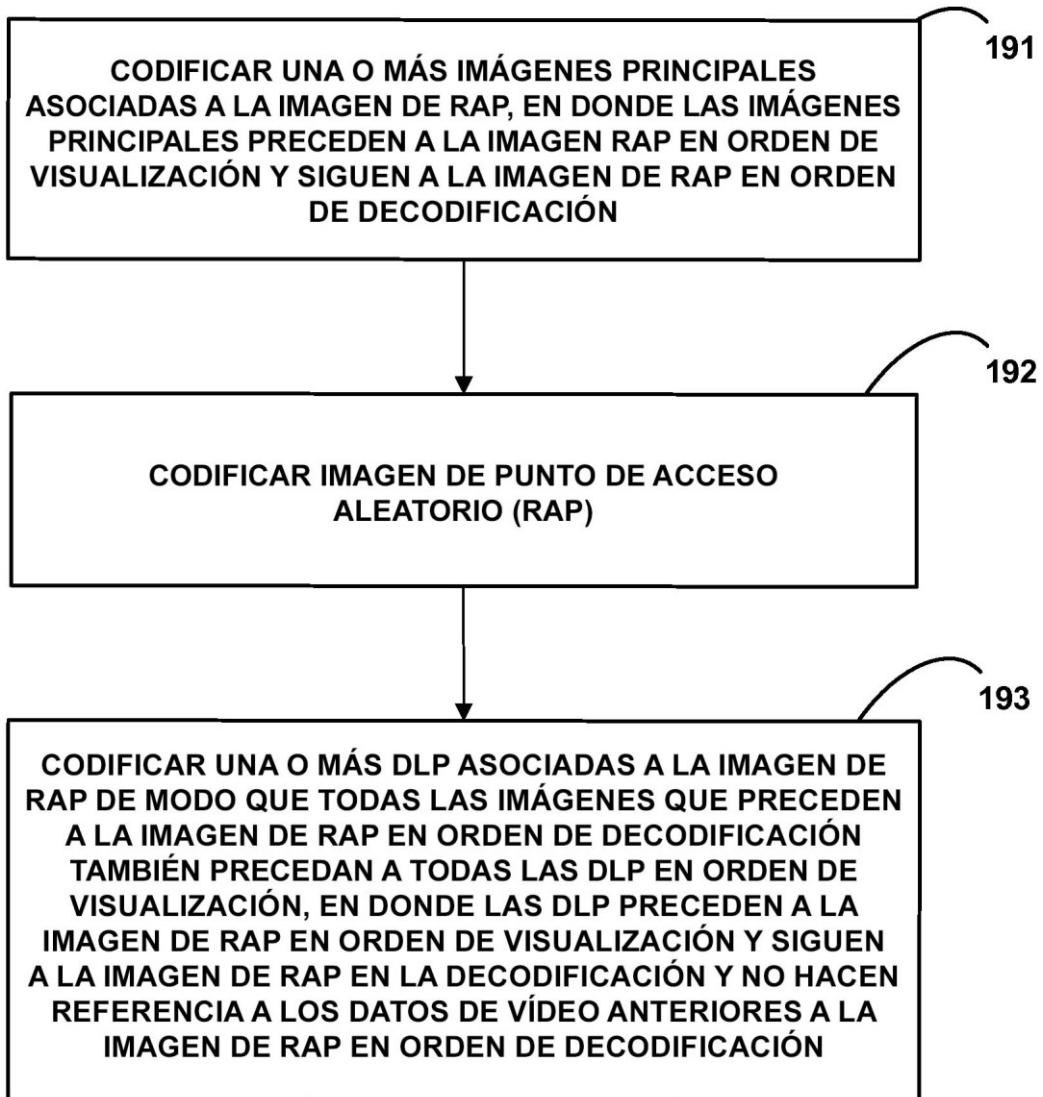


**FIG. 7**



**FIG. 8**





**FIG. 9**

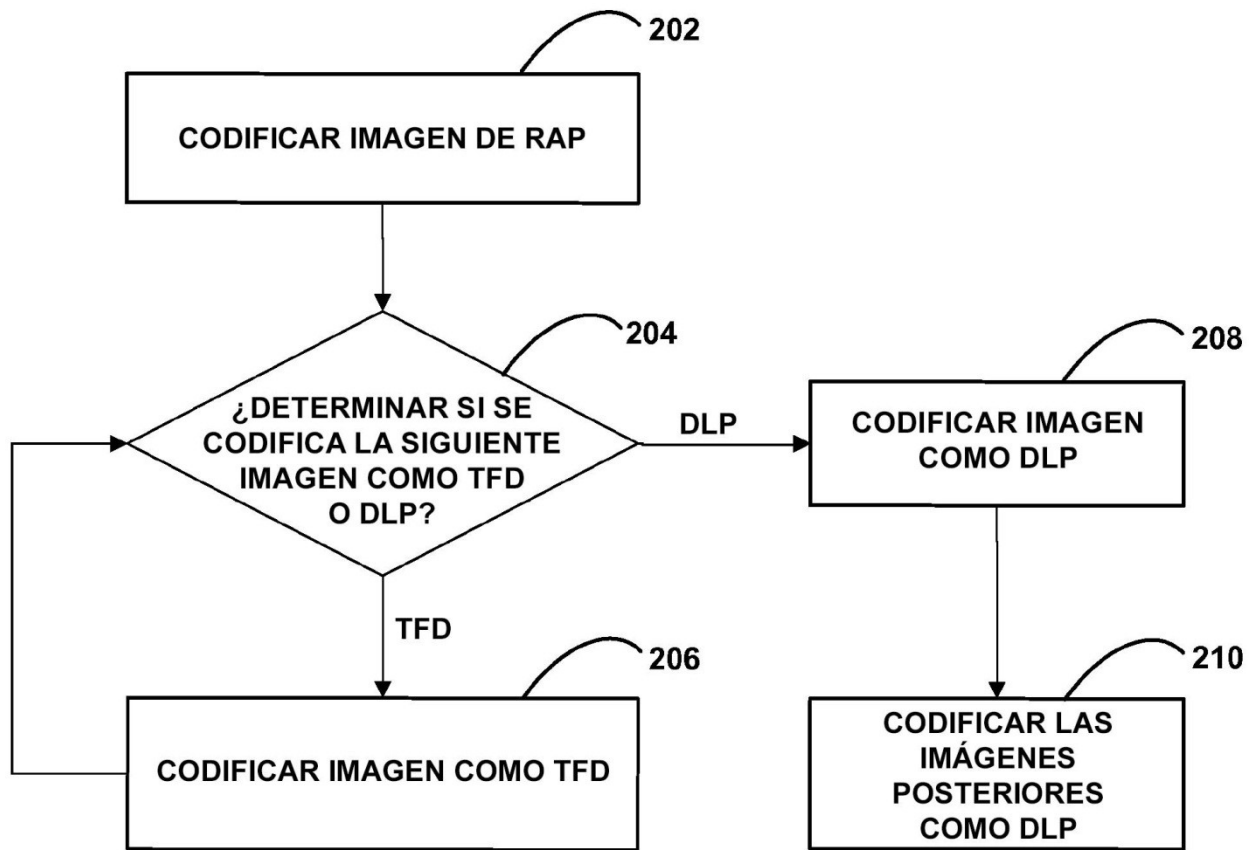


FIG. 10

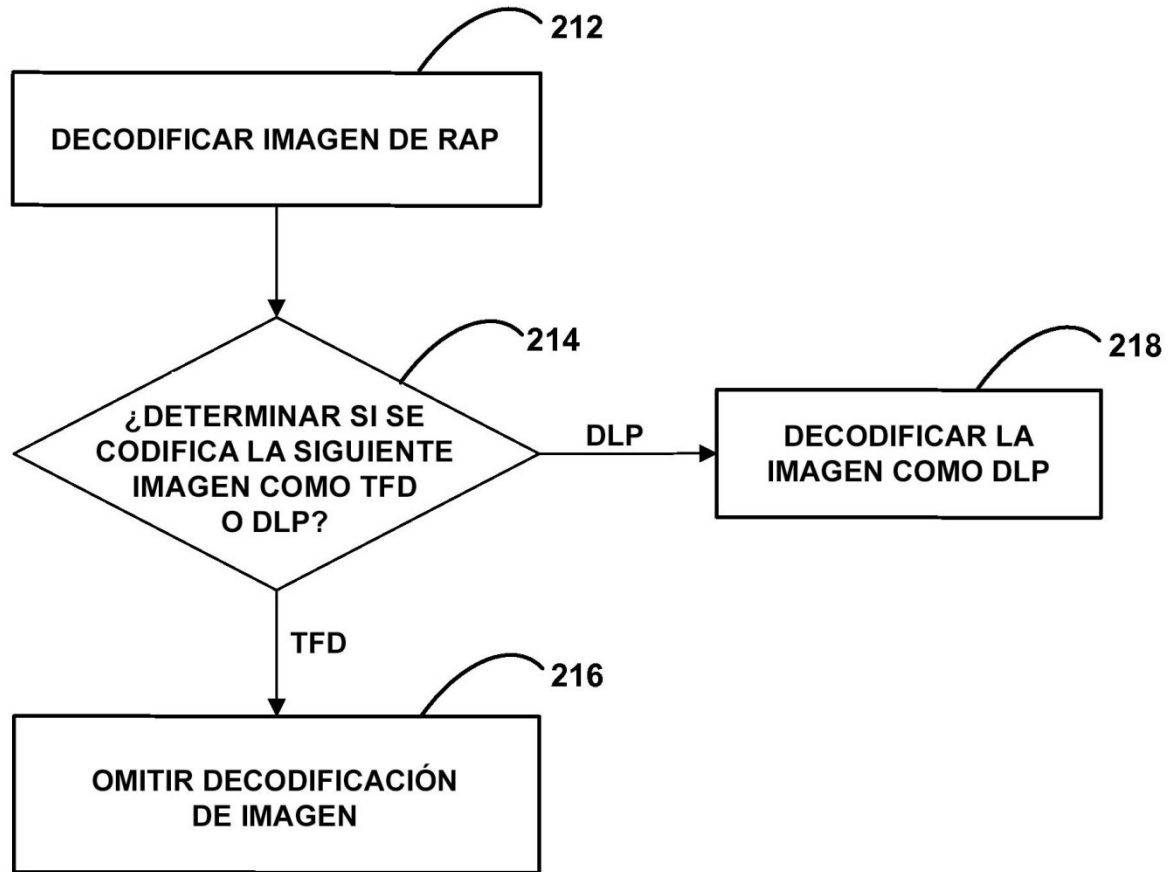


FIG. 11