

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 650 729**

51 Int. Cl.:

**H04N 19/597** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **20.12.2012 PCT/US2012/071049**

87 Fecha y número de publicación internacional: **27.06.2013 WO13096674**

96 Fecha de presentación y número de la solicitud europea: **20.12.2012 E 12816410 (0)**

97 Fecha y número de publicación de la concesión europea: **20.09.2017 EP 2795902**

54 Título: **Construcción de la lista de imágenes de referencia para codificación de vídeo de múltiples visualizaciones y tridimensional**

30 Prioridad:

**20.12.2011 US 201161578178 P**  
**19.12.2012 US 201213720622**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**22.01.2018**

73 Titular/es:

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

72 Inventor/es:

**CHEN, YING;**  
**WANG, YE-KUI y**  
**ZHANG, LI**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

**ES 2 650 729 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**

Construcción de la lista de imágenes de referencia para codificación de vídeo de múltiples visualizaciones y tridimensional

5

**CAMPO TÉCNICO**

[1] Esta divulgación se refiere a codificación de vídeo (es decir, codificación y/o descodificación de datos de vídeo.

10

**ANTECEDENTES**

[2] Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión por flujo de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC) (H.264/AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC) actualmente en desarrollo y las extensiones de tales normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de compresión de vídeo.

15

20

25

30

[3] Las técnicas de compresión de vídeo realizan la predicción espacial (entre imágenes) 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 (es decir, una trama de vídeo o una parte de una trama de vídeo) puede dividirse en bloques de vídeo. Los bloques de vídeo de un fragmento intra-codificado (I) de una imagen se codifican mediante predicción espacial con respecto a unas muestras de referencia de bloques próximos de la misma imagen. Los bloques de vídeo de un fragmento inter-codificado (P o B) de una imagen pueden usar predicción espacial con respecto a unas muestras de referencia de bloques próximos de la misma imagen, o la predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes pueden denominarse tramas y las imágenes de referencia pueden denominarse tramas de referencia.

35

40

45

[4] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque inter-codificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intra-codificado se codifica de acuerdo con un modo de intra-codificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio del píxel a un dominio de transformación, dando como resultado coeficientes residuales, los cuales pueden cuantificarse posteriormente. Los coeficientes cuantificados, inicialmente dispuestos en una matriz bidimensional, pueden explorarse con el fin de producir un vector unidimensional de coeficientes, y puede aplicarse la codificación por entropía para lograr aún más compresión.

50

55

[5] El documento WO 2011/074153 describe un aparato de descodificación de imagen que de acuerdo con un aspecto incluye: una unidad de almacenamiento de información de gestión que almacena la primera información de gestión para identificar áreas en cada una de las cuales se almacena una de las primeras señales de imagen descodificadas y segunda información de gestión para identificar áreas en cada una de las cuales se almacena una de las segundas señales de imagen descodificadas; y una unidad de control que notifica, haciendo referencia a la primera información de gestión, a una unidad de descodificación de imágenes un área en la que se almacena una imagen de referencia a la que se hace referencia cuando se descodifica una primera señal codificada, y notifica, mediante referencia a la primera información de gestión y la segunda información de gestión, a la unidad de descodificación de imágenes un área en la que se almacena una imagen de referencia a la que se hace referencia cuando se descodifica una segunda señal codificada.

**RESUMEN**

60

65

[6] La invención está definida en las páginas adjuntas, a las cuales debería hacerse referencia ahora. Un codificador de vídeo genera, basándose en un conjunto de imágenes de referencia de un componente de visualización actual, una lista de imágenes de referencia para el componente de visualización actual. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones. La lista de imágenes de referencia entre visualizaciones incluye imágenes de referencia que pertenecen a la misma unidad de acceso que el componente de visualización actual, pero están asociadas a diferentes visualizaciones que el componente de visualización actual. Inter-predicción o predicción entre visualizaciones de usuario del codificador de vídeo para codificar el componente de visualización actual basándose en una o más imágenes de referencia en la lista de

imágenes de referencia. Además, el codificador de vídeo genera un flujo de bits que incluye elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual. Un descodificador de vídeo analiza, desde el flujo de bits, los elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual. El descodificador de vídeo genera, basándose en el conjunto de imágenes de referencia, la lista de imágenes de referencia para el componente de visualización actual. Además, el descodificador de vídeo descodifica la visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia.

**[7]** En un aspecto, esta divulgación describe un procedimiento para descodificar vídeo de múltiples visualizaciones. El procedimiento comprende analizar, a partir de un flujo de bits, elementos sintácticos que indican un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones. El procedimiento también comprende generar, basándose en el conjunto de imágenes de referencia, una lista de imágenes de referencia para el componente de visualización actual. Además, el procedimiento comprende descodificar al menos una parte del componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia.

**[8]** En otro aspecto, esta divulgación describe un procedimiento para codificación de vídeo. El procedimiento comprende generar, basándose en un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso, una lista de imágenes de referencia para el componente de visualización actual. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones. El procedimiento también comprende codificar el componente de visualización actual basándose al menos en parte en una o más imágenes de referencia en la lista de imágenes de referencia. Además, el procedimiento comprende generar un flujo de bits que incluye elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual.

**[9]** En otro aspecto, esta divulgación describe un dispositivo de descodificación de vídeo que comprende uno o más procesadores configurados para analizar, a partir de un flujo de bits, elementos sintácticos que indican un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones. El uno o más procesadores también están configurados para generar, basándose en el conjunto de imágenes de referencia, una lista de imágenes de referencia para el componente de visualización actual. Además, el uno o más procesadores están configurados para descodificar al menos una parte del componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia.

**[10]** En otro aspecto, esta divulgación describe un dispositivo de codificación de vídeo que comprende uno o más procesadores configurados para generar, basándose en un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso, una lista de imágenes de referencia para el componente de visualización actual. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones. El uno o más procesadores también están configurados para codificar el componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia. Además, el uno o más procesadores están configurados para generar un flujo de bits que incluye elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual.

**[11]** En otro aspecto, esta divulgación describe un dispositivo de descodificación de vídeo que comprende medios para analizar, a partir de un flujo de bits, elementos sintácticos que indican un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones. El dispositivo de descodificación de vídeo también comprende medios para generar, basándose en el conjunto de imágenes de referencia, una lista de imágenes de referencia para el componente de visualización actual. Además, el dispositivo de descodificación de vídeo comprende medios para descodificar al menos una parte del componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia.

**[12]** En otro aspecto, esta divulgación describe un dispositivo de codificación de vídeo que comprende medios para generar, basándose en un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso, una lista de imágenes de referencia para el componente de visualización actual. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con

diferentes visualizaciones. El dispositivo de codificación de vídeo también comprende medios para codificar el componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia. Además, el dispositivo de codificación de vídeo comprende medios para generar un flujo de bits que incluye elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual.

[13] En otro aspecto, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones ejecutables por ordenador que, cuando son ejecutadas por uno o más procesadores de un dispositivo de decodificación de vídeo, configuran el dispositivo de decodificación de vídeo para analizar, a partir de un flujo de bits, elementos sintácticos que indican conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones. Las instrucciones también configuran el dispositivo de decodificación de vídeo para generar, basándose en el conjunto de imágenes de referencia, una lista de imágenes de referencia para el componente de visualización actual. Además, las instrucciones configuran el dispositivo de decodificación de vídeo para decodificar al menos una parte del componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia.

[14] En otro aspecto, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones ejecutables por ordenador que, cuando son ejecutadas por uno o más procesadores de un dispositivo de decodificación de vídeo, configuran el dispositivo de codificación de vídeo para generar, basándose en un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso, una lista de imágenes de referencia para el componente de visualización actual. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones. Las instrucciones también configuran el dispositivo de codificación de vídeo para codificar el componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia. Además, las instrucciones configuran el dispositivo de codificación de vídeo para generar un flujo de bits que incluye elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual.

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

## BREVE DESCRIPCIÓN DE LOS DIBUJOS

### [16]

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

La FIG. 2 es un diagrama conceptual que ilustra un ejemplo de orden de decodificación de codificación de vídeo de múltiples visualizaciones (MVC).

La FIG. 3 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción entre visualizaciones y temporal de MVC.

La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 5 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de operación realizada por el codificador de vídeo como parte de un proceso para codificar datos de vídeo, de acuerdo con una o más técnicas de esta divulgación.

La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de operación realizada por el decodificador de vídeo como parte de un proceso para decodificar datos de vídeo codificado, de acuerdo con una o más técnicas de esta divulgación.

## DESCRIPCIÓN DETALLADA

[17] Como parte de un proceso de inter-predicción, un codificador de vídeo señala, en un flujo de bits, un conjunto de imágenes de referencia de un componente de visualización particular. El conjunto de imágenes de referencia del componente de visualización particular puede incluir imágenes de referencia que están disponibles

para usarse para la inter-predicción de bloques dentro del componente de visualización particular. En la Codificación de Múltiples Visualizaciones (MVC) y la codificación de vídeo tridimensional (3DV), el codificador de vídeo puede usar componentes de visualización de la misma unidad de acceso que el componente de visualización particular como imágenes de referencia para codificar el componente de visualización particular. Sin embargo, el codificador de vídeo no señala tales imágenes de referencia entre visualizaciones en el conjunto de imágenes de referencia del componente de visualización. Esto puede disminuir la eficiencia de un descodificador de vídeo que descodifica el flujo de bits.

**[18]** De acuerdo con las técnicas de esta divulgación, un codificador de vídeo puede generar, basándose en un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso, una lista de imágenes de referencia para el componente de visualización actual. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso. Cada componente de visualización en el conjunto de imágenes de referencia entre visualizaciones puede estar asociado con una visualización diferente. Además, el codificador de vídeo puede codificar al menos partes del componente de visualización actual basándose al menos en parte en una o más imágenes de referencia en la lista de imágenes de referencia. Además, el codificador de vídeo puede generar un flujo de bits que incluye elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual.

**[19]** Asimismo, de acuerdo con las técnicas de esta divulgación, un descodificador de vídeo puede analizar, a partir de un flujo de bits, elementos sintácticos que indican un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso. El conjunto de imágenes de referencia incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización. Los componentes de visualización del conjunto de imágenes de referencia entre visualizaciones pertenecen a la unidad de acceso. Cada componente de visualización del conjunto de imágenes de referencia entre visualizaciones está asociado con una visualización diferente. El descodificador de vídeo puede generar, basándose en el conjunto de imágenes de referencia, una lista de imágenes de referencia para el componente de visualización actual. Además, el descodificador de vídeo puede descodificar al menos partes del componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia.

**[20]** Los dibujos adjuntos ilustran ejemplos. Los elementos indicados mediante números de referencia en los dibujos adjuntos corresponden a elementos indicados mediante números de referencia similares en la siguiente descripción. En esta divulgación, los elementos que tienen nombres que comienzan con palabras ordinales (por ejemplo, "primero", "segundo", "tercero", etc.) no necesariamente implican que los elementos tienen un orden particular. Más bien, tales palabras ordinales se usan simplemente para referirse a diferentes elementos de un mismo o similar tipo.

**[21]** La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo de ejemplo 10 que puede utilizar las técnicas de esta divulgación. Tal como se utiliza descrito en el presente documento, el término "codificador de vídeo" se refiere genéricamente tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" se refieren genéricamente a la codificación de vídeo o la descodificación de vídeo.

**[22]** Como se muestra en la FIG. 1, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 y un dispositivo de destino 14. El dispositivo de origen 12 genera datos de vídeo codificados. En consecuencia, el dispositivo de origen 12 puede denominarse dispositivo de codificación de vídeo o aparato de codificación de vídeo. El dispositivo de destino 14 descodifica los datos de vídeo codificados generados por el dispositivo de origen 12. En consecuencia, el dispositivo de destino 14 puede denominarse dispositivo de descodificación de vídeo o aparato de descodificación de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son ejemplos de dispositivos de codificación de vídeo o aparatos de codificación de vídeo.

**[23]** El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia variedad de dispositivos, incluyendo ordenadores de sobremesa, dispositivos informáticos móviles, ordenadores plegables (es decir, portátiles), ordenadores de tableta, descodificadores, equipos telefónicos portátiles tales como los denominados teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores de coche o similares.

**[24]** El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 a través de un canal 16. El canal 16 puede comprender uno o más medios o dispositivos capaces de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender uno o más medios de comunicaciones que permiten al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de origen 12 puede modular los datos de vídeo codificados de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. El uno o más medios de comunicaciones pueden incluir medios de comunicaciones inalámbricos y/o cableados, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El uno o más medios de

comunicación pueden formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global (por ejemplo, Internet). El uno o más medios de comunicación pueden incluir routers, conmutadores, estaciones base u otros equipos que faciliten la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

**[25]** En otro ejemplo, el canal 16 puede incluir un medio de almacenamiento que almacena datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento a través del acceso al disco o el acceso a la tarjeta. El medio de almacenamiento puede incluir una variedad de medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria flash u otros medios adecuados de almacenamiento digital para almacenar datos de vídeo codificados.

**[26]** En un ejemplo adicional, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacene los datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder a datos de vídeo codificados almacenados en el servidor de archivos u otro dispositivo de almacenamiento intermedio a través de transmisión o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir los datos de vídeo codificados al dispositivo de destino 14. Entre los ejemplos de servidores de archivos se incluyen servidores web (por ejemplo, para un sitio web), servidores de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados a la red (NAS) y unidades de disco local.

**[27]** El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de una conexión de datos estándar, como una conexión a Internet. Entre los ejemplos de tipos de conexiones de datos pueden incluirse canales inalámbricos (por ejemplo, conexiones Wi-Fi), conexiones alámbricas (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 servidor de archivos puede ser una transmisión sin descarga, una transmisión de descarga o una combinación de ambas.

**[28]** Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de una diversidad de aplicaciones multimedia, tales como radiodifusiones de televisión por el aire, radiodifusiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, mediante Internet, codificación de datos de vídeo para su almacenamiento en un medio de almacenamiento de datos, descodificación de datos de vídeo almacenados en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 puede configurarse para soportar transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tales como radiodifusión de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

**[29]** En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos ejemplos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. La fuente de vídeo 18 puede incluir un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene datos de vídeo previamente capturados, una interfaz de alimentación de vídeo para recibir datos de vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de vídeo, o una combinación de dichas fuentes de datos de vídeo.

**[30]** El codificador de vídeo 20 puede codificar datos de vídeo de la fuente de vídeo 18. En algunos ejemplos, el dispositivo de origen 12 directamente transmite los datos de vídeo codificados al dispositivo destino 14 a través de un canal de comunicación 22. En otros ejemplos, los datos de vídeo codificados también pueden almacenarse en un medio de almacenamiento o un servidor de archivos para un acceso posterior mediante el dispositivo de destino 14 para la descodificación y/o reproducción.

**[31]** En el ejemplo de la FIG. 1, el dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos ejemplos, la interfaz de entrada 28 incluye un receptor y/o un módem. La interfaz de entrada 28 puede recibir los datos de vídeo codificados por el canal 16. El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En general, el dispositivo de visualización 32 muestra datos de vídeo descodificados. El dispositivo de visualización 32 puede comprender una variedad de dispositivos de visualización, tales como 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.

**[32]** En algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 funcionan de acuerdo con una norma de compresión de vídeo, tal como la ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluidas sus extensiones de codificación de vídeo escalable (SVC) y codificación de vídeo de múltiples visualizaciones (MVC). Desde el 19 de diciembre de 2012, se puede descargar un borrador de la extensión de MVC de H.264 en [http://wftp3.itu.int/av-arch/jvt-site/2009\\_01\\_Geneva/JVT-AD007.zip](http://wftp3.itu.int/av-arch/jvt-site/2009_01_Geneva/JVT-AD007.zip), cuyo contenido completo se incorpora en el presente documento como referencia.

**[33]** En otros ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas de compresión de vídeo, incluyendo la norma de Codificación de Vídeo de Alta Eficiencia (HEVC) actualmente en desarrollo. Un borrador de la próxima norma HEVC, denominada "Borrador de Trabajo de HEVC 4", se describe en Bross et al., "WD4: Borrador de Trabajo 4 de Codificación de Vídeo de Alta Eficiencia", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 6.<sup>a</sup> reunión: Turín, Italia, julio de 2011, que desde el 19 de diciembre de 2012 puede descargarse desde: [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/6\\_Torino/wg11/JCTVC-F803-v8.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F803-v8.zip), cuyo contenido completo se incorpora en el presente documento como referencia. Otro borrador de la siguiente norma HEVC, denominado "Borrador de Trabajo 9 de la HEVC" se describe en el documento de Bross et al. "Borrador 9 de Memoria Descriptiva Textual de Codificación de Vídeo de Alta Eficacia (HEVC)", Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 11.<sup>a</sup> reunión: Shanghái, China, octubre de 2012, que, desde el 19 de diciembre de 2012, puede descargarse [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/11\\_Shanghai/wg11/JCTVC-K1003-v8.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v8.zip), cuyo contenido completo se incorpora en el presente documento como referencia. Además, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una extensión de múltiples visualizaciones de HEVC. Un borrador de la extensión de múltiples visualizaciones de HEVC se describe en Tech et al., "Borrador de Trabajo 2 de MV-HEVC", Equipo de Colaboración Conjunta en Desarrollo de Extensión de Codificación de Vídeo 3D de ITU-T SG16 WP3 e ISO/IEC JTC 1/SC 29/WG11, 2.<sup>a</sup> reunión: Shanghái, China, octubre de 2012, que desde el 19 de diciembre de 2012 se puede descargar desde: [http://phenix.int-evry.fr/jct2/doc\\_end\\_user/documents/2\\_Shanghai/wg11/JCT3V-B1004-v1.zip](http://phenix.int-evry.fr/jct2/doc_end_user/documents/2_Shanghai/wg11/JCT3V-B1004-v1.zip), cuyo contenido completo se incorpora en el presente documento como referencia. Además, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una extensión 3DV de HEVC. Un borrador de la extensión 3DV de HEVC se describe en Hannuksela et al., "Modelo de Pruebas 1 de 3D-HEVC", Equipo de Colaboración Conjunta en Desarrollo de Extensión de Codificación de Vídeo 3D de ITU-T SG16 WP3 e ISO/IEC JTC 1/SC 29/WG11, 1.<sup>a</sup> reunión: Estocolmo, Suecia, julio de 2012, que desde el 19 de diciembre de 2012 puede descargarse desde [http://phenix.it-sudparis.eu/jct3v/doc\\_end\\_user/documents/1\\_Stockholm/wg11/JCT3V-A1005-v1.zip](http://phenix.it-sudparis.eu/jct3v/doc_end_user/documents/1_Stockholm/wg11/JCT3V-A1005-v1.zip), cuyo contenido completo se incorpora en el presente documento como referencia. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna técnica ni norma de codificación particular.

**[34]** La FIG. 1 es meramente un ejemplo y las técnicas de esta divulgación pueden aplicarse a ajustes de codificación de vídeo (por ejemplo, codificación de vídeo o descodificación de vídeo) que no incluyen necesariamente ninguna comunicación de datos entre dispositivos de codificación y descodificación. En otros ejemplos, los datos se recuperan de una memoria local, se transmiten a través de una red o similar. Un dispositivo de codificación puede codificar y almacenar datos en la memoria, y/o un dispositivo de descodificación puede recuperar y descodificar datos de la memoria. En muchos ejemplos, la codificación y descodificación se realiza mediante dispositivos que no se comunican entre sí, sino que simplemente codifican datos en la memoria y/o recuperan y descodifican datos de la memoria.

**[35]** El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera entre una variedad de circuitos adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), conjuntos de compuertas programables sobre el terreno (FPGA), lógica discreta, hardware o cualquier combinación de los mismos. Si las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio de almacenamiento legible por ordenador no transitorio adecuado, y puede ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cualquiera de los anteriores (incluyendo hardware, software, una combinación de hardware y software, etc.) puede considerarse como uno o más procesadores. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

**[36]** Esta divulgación puede referirse en general al codificador de vídeo 20 que "señala" cierta información a otro dispositivo, tal como el descodificador de vídeo 30. El término "señalar" puede referirse en general a la comunicación de elementos sintácticos y/u otros datos usados para descodificar los datos de vídeo comprimidos. Dicha comunicación puede producirse en tiempo real o casi real. De forma alternativa, dicha comunicación puede producirse durante un lapso de tiempo, tal como podría ocurrir cuando se almacenan elementos sintácticos en un medio de almacenamiento legible por ordenador en un flujo de bits codificado en el momento de la codificación, que a continuación un dispositivo de descodificación puede recuperar en cualquier momento tras haber sido almacenado en este medio.

**[37]** Como se ha mencionado brevemente anteriormente, el codificador de vídeo 20 codifica los datos de vídeo. Los datos de vídeo pueden comprender una o más imágenes. Cada una de las imágenes puede ser una imagen fija. En algunos casos, una imagen puede denominarse una "trama" de vídeo. El codificador de vídeo 20 puede generar un flujo de bits que incluye una secuencia de bits que forman una representación codificada de los datos de vídeo. La representación codificada de los datos de vídeo puede incluir imágenes codificadas y datos asociados. Una imagen codificada es una representación codificada de una imagen. Los datos asociados pueden incluir conjuntos de parámetros de secuencia (SPS), conjuntos de parámetros de imagen (PPS) y otras estructuras sintácticas. Un SPS puede contener parámetros aplicables a cero o más secuencias de imágenes. Un PPS puede contener

parámetros aplicables a cero o más imágenes. Una estructura sintáctica puede ser un conjunto de cero o más elementos sintácticos presentes conjuntamente en un flujo de bits en un orden específico.

**[38]** Un codificador de vídeo 20 actúa típicamente sobre bloques de vídeo de tramas de vídeo individuales con el fin de codificar los datos de vídeo. En H.264/AVC, un bloque de vídeo puede corresponder a un macro-bloque o una división de un macro-bloque en H.264/AVC. En HEVC, un bloque de vídeo puede corresponder a una unidad de codificación (CU). Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada. Además, cada trama de vídeo puede incluir una pluralidad de fragmentos. Cada fragmento puede incluir una pluralidad de bloques de vídeo.

**[39]** Cuando el codificador de vídeo 20 codifica un bloque de vídeo, el codificador de vídeo 20 puede generar un bloque de píxeles predictivo que se corresponda con el bloque de vídeo. El codificador de vídeo 20 puede realizar intra-predicción o inter-predicción para generar los bloques de píxeles predictivos. Cuando el codificador de vídeo 20 realiza la intra-predicción en un bloque de vídeo, el codificador de vídeo 20 puede generar, basándose en muestras (por ejemplo, valores de componentes de píxeles) de la misma trama de vídeo que el bloque de vídeo, un bloque de píxeles predictivo que se corresponde con el bloque de vídeo. Cuando el codificador de vídeo 20 realiza inter-predicción para generar un bloque de píxeles predictivo que se corresponde con un bloque de vídeo, el codificador de vídeo 20 puede generar el bloque de píxeles predictivo basándose en muestras de una o más imágenes de referencia. Las imágenes de referencia pueden ser imágenes distintas de la imagen que contiene el bloque de vídeo.

**[40]** Después de que el codificador de vídeo 20 genere un bloque de píxeles predictivo que corresponde a un bloque de vídeo, el codificador de vídeo 20 puede generar un bloque de vídeo residual que corresponde al bloque de vídeo. Cada muestra en el bloque de vídeo residual puede basarse en una diferencia entre las muestras correspondientes en el bloque de vídeo y el bloque de píxeles predictivo. El codificador de vídeo 20 puede aplicar una transformación al bloque de vídeo residual para generar uno o más bloques de coeficientes. El codificador de vídeo 20 puede aplicar varias transformaciones al bloque de vídeo residual. Por ejemplo, el codificador de vídeo 20 puede aplicar una transformación tal como una transformación de coseno discreta (DCT), una transformación de número entero, una transformación de wavelet o una transformación conceptualmente similar al bloque de vídeo residual. El codificador de vídeo 20 puede cuantificar el bloque de coeficientes para reducir aún más el número de bits utilizados para representar el bloque de vídeo. Después de cuantificar el bloque de coeficientes, el codificador de vídeo 20 puede realizar y codificar por entropía (por ejemplo, codificación aritmética binaria adaptable al contexto (CABAC), codificación de longitud variable adaptable al contexto (CAVLC), codificación Golomb exponencial, etc.) elementos sintácticos que representan coeficientes en el bloque de coeficientes y en otros elementos sintácticos asociados con el bloque de vídeo. El codificador de vídeo 20 puede generar un flujo de bits que incluye los elementos sintácticos codificados por entropía asociados con el bloque de vídeo.

**[41]** El descodificador de vídeo 30 puede recibir un flujo de bits que incluye una representación codificada de datos de vídeo. Por ejemplo, el descodificador de vídeo 30 puede analizar el flujo de bits para extraer elementos sintácticos del flujo de bits. Como parte de la extracción de los elementos sintácticos del flujo de bits, el descodificador de vídeo 30 puede aplicar la descodificación por entropía a partes del flujo de bits. Para cada bloque de vídeo respectivo de cada trama de datos de vídeo, el descodificador de vídeo 30 puede realizar, basándose al menos en parte en los elementos sintácticos asociados con el respectivo bloque de vídeo, inter- o intra-predicción para generar un bloque de píxeles predictivo para el bloque de vídeo respectivo. Además, el descodificador de vídeo 30 puede cuantificar inversamente coeficientes de bloques de coeficientes asociados con el bloque de vídeo respectivo y puede aplicar una o más transformaciones inversas a los bloques de coeficientes para generar un bloque de vídeo residual para el bloque de vídeo respectivo. El descodificador de vídeo 30 puede entonces reconstruir el bloque de vídeo respectivo basándose, al menos en parte, en el bloque de vídeo residual y el bloque de píxeles predictivo. De este modo, mediante la reconstrucción de cada uno de los bloques de vídeo de una trama, el descodificador de vídeo 30 puede reconstruir la trama.

**[42]** Como se ha mencionado anteriormente, el codificador de vídeo 20 puede llevar a cabo la inter-predicción para generar un bloque de píxeles predictivo que corresponde a un bloque de vídeo particular. Más específicamente, el codificador de vídeo 20 puede realizar una inter-predicción unidireccional o una inter-predicción bidireccional para generar el bloque de píxeles predictivo.

**[43]** Cuando el codificador de vídeo 20 realiza la inter-predicción unidireccional para un bloque de vídeo particular, el codificador de vídeo 20 puede buscar un bloque de referencia en las imágenes de referencia de una sola lista de imágenes de referencia. El bloque de referencia puede ser un bloque de muestras que es similar al bloque de vídeo particular. Además, cuando el codificador de vídeo 20 realiza una inter-predicción unidireccional, el codificador de vídeo 20 puede generar información de movimiento para el bloque de vídeo particular. La información de movimiento para el bloque de vídeo particular puede incluir un vector de movimiento y un índice de imagen de referencia. El vector de movimiento puede indicar un desplazamiento espacial entre una posición del bloque de vídeo particular en la trama actual (es decir, la trama que incluye el bloque de vídeo particular) y una posición del bloque de referencia en la trama de referencia. El índice de imagen de referencia indica una posición en la lista de imágenes de referencia de la trama de referencia que contiene la lista de imágenes de referencia. El bloque de píxeles predictivo para el bloque de vídeo actual pueden ser iguales a unas muestras correspondientes del bloque

de referencia.

**[44]** Cuando el codificador de vídeo 20 realiza la inter-predicción bidireccional para un bloque de vídeo particular, el codificador de vídeo 20 puede buscar un primer bloque de referencia en las imágenes de referencia de una primera lista de imágenes de referencia ("lista 0") y puede buscar un segundo bloque de referencia en las imágenes de referencia de una segunda lista de imágenes de referencia ("lista 1"). El codificador de vídeo 20 puede generar, basándose al menos en parte en el primer y el segundo bloques de referencia, el bloque de píxeles predictivo para el bloque de vídeo particular. Además, el codificador de vídeo 20 puede generar un primer vector de movimiento que indica un desplazamiento espacial entre el bloque de vídeo particular y el primer bloque de referencia. El codificador de vídeo 20 también puede generar un primer índice de imagen de referencia que identifica una ubicación en la primera lista de imágenes de referencia de la imagen de referencia que contiene el primer bloque de referencia. Además, el codificador de vídeo 20 puede generar un segundo vector de movimiento que indica un desplazamiento espacial entre el bloque de vídeo particular y el segundo bloque de referencia. El codificador de vídeo 20 también puede generar un segundo índice de imagen de referencia que identifica una ubicación en la segunda lista de imágenes de referencia de la imagen de referencia que incluye el segundo bloque de referencia.

**[45]** Cuando el codificador de vídeo 20 realiza la inter-predicción unidireccional en un bloque de vídeo particular, el descodificador de vídeo 30 puede utilizar la información de movimiento del bloque de vídeo particular para identificar la muestra de referencia del bloque de vídeo particular. El descodificador de vídeo 30 puede generar entonces el bloque de píxeles predictivo del bloque de vídeo particular basándose en la muestra de referencia del bloque de vídeo particular. Cuando el codificador de vídeo 20 realiza la inter-predicción bidireccional en un bloque de vídeo particular, el descodificador de vídeo 30 puede utilizar la información de movimiento del bloque de vídeo particular para identificar las dos muestras de referencia del bloque de vídeo particular. El descodificador de vídeo 30 puede generar el bloque de píxeles predictivo del bloque de vídeo particular basándose en las dos muestras de referencia del bloque de vídeo particular.

**[46]** H.264/AVC puede ampliarse para proporcionar Codificación de Vídeo de Múltiples Visualizaciones (MVC). En la extensión de MVC para H.264, puede haber múltiples visualizaciones de la misma escena desde diferentes puntos de vista. El término "unidad de acceso" se utiliza para referirse al conjunto de imágenes que corresponden a la misma instancia de tiempo. Así, los datos de vídeo pueden conceptualizarse como una serie de unidades de acceso que se producen a lo largo del tiempo.

**[47]** La FIG. 2 es un diagrama conceptual que ilustra un ejemplo de orden de descodificación de MVC. En el ejemplo de la FIG. 2 cada cuadrado corresponde a una imagen. Las columnas de cuadrados corresponden a las unidades de acceso. Cada unidad de acceso puede definirse para contener las imágenes codificadas de todas las visualizaciones de un instante de tiempo. Las filas de cuadrados corresponden a las visualizaciones. En el ejemplo de la FIG. 2, las unidades de acceso están etiquetadas como T0 ... T7 y las visualizaciones están etiquetadas como S0 ... S7. Debido a que cada imagen de una unidad de acceso se descodifica antes de cualquier componente de visualización de la siguiente unidad de acceso, el orden de descodificación de la FIG. 7 puede denominarse codificación de primera vez. Como se ilustra en el ejemplo de la FIG. 2, el orden de descodificación de las imágenes puede no ser idéntico al orden de salida o de visualización de las imágenes.

**[48]** La extensión de MVC de H.264/AVC, la extensión de MVC de HEVC y otras normas proporcionan predicción entre visualizaciones. La predicción entre visualizaciones es similar a la inter-predicción utilizada en H.264/AVC y puede usar los mismos elementos sintácticos. Sin embargo, cuando el codificador de vídeo 20 realiza la predicción entre visualizaciones en un bloque de vídeo, el codificador de vídeo 20 puede usar, como una imagen de referencia, una imagen que está en la misma unidad de acceso que el bloque de vídeo, pero en una visualización diferente. En contraste, la inter-predicción convencional solo usa imágenes en diferentes unidades de acceso como imágenes de referencia. En MVC, una visualización se denomina "visualización básica" si un descodificador de vídeo (por ejemplo, descodificador de vídeo 30) puede descodificar imágenes de la visualización sin referencia a imágenes de ninguna otra visualización. La realización de compensación de movimiento en una imagen en una visualización diferente de una misma unidad de acceso puede denominarse compensación de movimiento de disparidad. Esta divulgación puede referirse a un componente de visualización usado como una imagen de referencia para la predicción entre visualizaciones como un componente de visualización de referencia entre visualizaciones. En algunos ejemplos, una imagen de referencia entre visualizaciones puede ser un componente de visualización con `nal_ref_flag` igual a 1 o puede ser un componente de visualización con `nal_ref_flag` igual a 0.

**[49]** La FIG. 3 es un diagrama conceptual que ilustra un ejemplo de estructura de predicción entre visualizaciones y temporal de MVC. En el ejemplo de la FIG. 3 cada cuadrado corresponde a una imagen. Los cuadrados etiquetados como "I" son imágenes intra-predichas. Los cuadrados etiquetados como "P" son imágenes inter-predichas unidireccionalmente. Los cuadrados etiquetados como "B" y "b" son imágenes inter-predichas bidireccionalmente. Los cuadrados etiquetados como "b" son imágenes predichas bidireccionalmente que no se utilizan como imágenes de referencia para otras imágenes. Los cuadrados etiquetados como "B" son imágenes predichas bidireccionalmente que pueden usarse como imágenes de referencia para otras imágenes. Una flecha que apunta desde un primer cuadrado a un segundo cuadrado indica que el primer cuadrado está disponible en la inter-

predicción como imagen de referencia para el segundo cuadrado. Como se indica mediante las flechas verticales en la FIG. 3, las imágenes en diferentes visualizaciones de la misma unidad de acceso pueden estar disponibles como imágenes de referencia. El uso de una imagen de una unidad de acceso como imagen de referencia para otra imagen de la misma unidad de acceso puede denominarse predicción entre visualizaciones.

5 **[50]** En la extensión de MVC de H.264/AVC, la predicción entre visualizaciones puede realizarse como si la imagen en otra visualización fuera una imagen de referencia de inter-predicción. Cuando se codifica una imagen actual en una visualización no básica, se puede añadir una imagen a una lista de imágenes de referencia si la imagen está en una visualización diferente pero con una misma instancia temporal (es decir, unidad de acceso) que la imagen actual. Una imagen de referencia de predicción entre visualizaciones puede disponerse en cualquier posición de una lista de imágenes de referencia, como cualquier otro tipo de imagen de referencia de inter-predicción. En la extensión de MVC de H.264/AVC, el codificador de vídeo 20 puede señalar, en una extensión de un SPS para una secuencia de vídeo, las imágenes de referencia entre visualizaciones que potencialmente pueden usarse para imágenes en la secuencia de vídeo. La Tabla 1 siguiente es un ejemplo de sintaxis para la extensión del SPS.

TABLA 1

| seq_parameter_set_mvc_extension(){                                | Descriptor |
|---|------------|
| <b>num_views_minus1</b>   | ue(v)      |
| for(i = 0; i <= num_views_minus1; i++)                            |            |
| <b>view_id[i]</b>   | ue(v)      |
| for(i = 1; i <= num_views_minus1; i++){                           |            |
| <b>num_anchor_refs_I0[i]</b>                                      | ue(v)      |
| for(j = 0; j < num_anchor_refs_I0[i]; j++)                        |            |
| <b>anchor_ref_I0[i][j]</b>  | ue(v)      |
| <b>num_anchor_refs_I1[i]</b>                                      | ue(v)      |
| for(j = 0; j < num_anchor_refs_I1[i]; j++)                        |            |
| <b>anchor_ref_I1[i][j]</b>  | ue(v)      |
| }   |            |
| for(i = 1; i <= num_views_minus1; i++){                           |            |
| <b>num_non_anchor_refs_I0[i]</b>                                  | ue(v)      |
| for(j = 0; j < num_non_anchor_refs_I0[i]; j++)                    |            |
| <b>non_anchor_ref_I0[i][j]</b>                                    | ue(v)      |
| <b>num_non_anchor_refs_I1[i]</b>                                  | ue(v)      |
| for(j = 0; j < num_non_anchor_refs_I1[i]; j++)                    |            |
| <b>non_anchor_ref_I1[i][j]</b>                                    | ue(v)      |
| }   |            |
| <b>num_level_values_signalled_minus1</b>                          | ue(v)      |
| for(i = 0; i <= num_level_values_signalled_minus1; i++){          |            |
| <b>level_idc[i]</b>   | u(8)       |
| <b>num_applicable_ops_minus1[i]</b>                               | ue(v)      |
| for(j = 0; j <= num_applicable_ops_minus1[i]; j++) {              |            |
| <b>applicable_op_temporal_id[i][j]</b>                            | u(3)       |
| <b>applicable_op_num_target_views_minus1[i][j]</b>                | ue(v)      |
| for(k = 0; k <= applicable_op_num_target_views_minus1[i][j]; k++) |            |
| <b>applicable_op_target_view-id[i][j][k]</b>                      | ue(v)      |
| <b>applicable_op_num_views_minus1[i][j]</b>                       | ue(v)      |
| }   |            |
| }   |            |
| }   |            |

[51] En la Tabla 1 y en las tablas siguientes, los elementos sintácticos con descriptor de tipo ue(v) son números enteros de longitud variable sin signo codificados utilizando codificación Golomb exponencial de orden 0 con el bit izquierdo primero. En la Tabla 1 y en las siguientes tablas, un elemento sintáctico que tiene un descriptor de tipo de la forma u(n), donde n es un número entero no negativo, son valores sin signo de longitud n. Por ejemplo, los elementos sintácticos con los descriptores de tipo u(3) y u(8) son un número entero sin signo con 3 y 8 bits, respectivamente.

[52] En la extensión de SPS MVC de la Tabla 1, el elemento sintáctico "num\_views\_minus1", más 1, indica el número máximo de visualizaciones codificadas en una secuencia de vídeo codificada. Los elementos sintácticos "view\_id[i]" indican identificadores de visualización de componentes de visualización con identificadores de orden de visualización iguales a i. Un identificador de visualización puede ser un valor que identifica una posición de cámara particular. Todos los componentes de visualización que pertenecen a la misma posición de cámara pueden estar asociados con el mismo identificador de visualización. Un índice de orden de visualización es un número entero con signo que especifica el orden de las visualizaciones codificadas de izquierda a derecha. Por el contrario, un identificador de visualización (viewId) no implica ningún orden de las posiciones de la cámara.

[53] Además, la extensión de SPS MVC puede incluir, para cada visualización, elementos sintácticos "anchor\_ref\_10" que indican identificadores de visualización de imágenes que el descodificador de vídeo 30 puede incluir en una versión inicial de la lista de imágenes de referencia 0 para la predicción entre visualizaciones de imágenes de anclaje. Una imagen de anclaje es una imagen codificada en la que todos los fragmentos solo hacen referencia a los fragmentos en la misma unidad de acceso. Es decir, el codificador de vídeo 20 puede usar la predicción entre visualizaciones para codificar una imagen de anclaje, pero el codificador de vídeo 20 no usa interpredicción para codificar la imagen de anclaje. La extensión de SPS MVC también puede incluir, para cada visualización, elementos sintácticos "anchor\_ref\_11" que indican identificadores de visualización de imágenes que el descodificador de vídeo 30 puede incluir en una versión inicial de la lista de imágenes de referencia 1 para la predicción entre visualizaciones de imágenes de anclaje. Además, la extensión de SPS MVC puede incluir, para cada visualización, elementos sintácticos "non\_anchor\_refs\_10" que indican identificadores de visualización de imágenes que el descodificador de vídeo 30 puede incluir en la lista 0 para la predicción entre visualizaciones de componentes de visualización que no son de anclaje. De manera similar, la extensión de SPS MVC puede incluir, para cada visualización, elementos sintácticos "non\_anchor\_refs\_11" que indican identificadores de visualización de imágenes que el descodificador de vídeo 30 puede incluir en la lista 1 para la predicción entre visualizaciones de componentes de visualización que no son de anclaje. De esta manera, la extensión de SPS MVC de la Tabla 1 puede proporcionar diferentes relaciones de predicción para las imágenes de anclaje y las imágenes que no son de anclaje de la misma visualización.

[54] En la extensión de MVC de H.264/AVC, una unidad NAL puede incluir una cabecera de unidad NAL de un byte y una extensión de cabecera de unidad MVC NAL de tres bytes si el tipo de unidad NAL es una unidad NAL de prefijo o una unidad NAL de capa de codificación de vídeo (VCL) MVC. Una unidad NAL de prefijo en MVC puede contener solo una cabecera de unidad NAL y la extensión de cabecera de unidad MVC NAL. Una unidad MVC VCL NAL puede ser una unidad NAL de fragmento codificado. En H.264/AVC, el término "unidad VCL NAL" es un término colectivo para referirse a unidades NAL de fragmento codificado y unidades NAL de división de datos de fragmento codificado. La Tabla 2 siguiente indica una sintaxis de ejemplo para la extensión de cabecera de unidad MVC NAL ("nal\_unit\_header\_mvc\_extension").

**TABLA 2**

| nal_unit_header_mvc_extension(){ | C     | Descriptor |
|----------------------------------|-------|------------|
| <b>non_idr_flag</b>              | Todos | u(1)       |
| <b>priority_id</b>               | Todos | u(6)       |
| <b>view_id</b>                   | Todos | u(10)      |
| <b>Temporal_id</b>               | Todos | u(3)       |
| <b>anchor_pic_flag</b>           | Todos | u(1)       |
| <b>inter_view_flag</b>           | Todos | u(1)       |
| <b>reserved_one_bit</b>          | Todos | u(1)       |
| }                                |       |            |

[55] En la sintaxis de ejemplo de la Tabla 2, el non\_idr\_flag indica si la unidad NAL pertenece a una unidad NAL de actualización de descodificación instantánea (IDR). Un punto de acceso aleatorio es una imagen que incluye solo 1 fragmentos. Un punto de acceso aleatorio es una imagen que incluye solo 1 fragmentos. El elemento sintáctico priority\_id se puede usar para adaptación de una vía, en el que la adaptación se puede hacer comprobando priority\_id. El elemento sintáctico view\_id puede indicar un identificador de visualización de una visualización actual.

La unidad NAL puede incluir una representación codificada de un fragmento de un componente de visualización de la visualización actual. El elemento sintáctico temporal\_id puede indicar un nivel temporal de la unidad NAL. El nivel temporal de una unidad NAL puede indicar una velocidad de imagen asociada con la unidad NAL. El elemento sintáctico anchor\_pic\_flag puede indicar si la unidad NAL pertenece a una imagen de anclaje que se puede usar para acceso aleatorio. El elemento sintáctico inter\_view\_flag indica si un componente de visualización actual se utiliza para predicción entre visualizaciones para unidades NAL en otras visualizaciones. La unidad NAL puede incluir una representación codificada de un fragmento de la imagen actual.

**[56]** Además de codificar múltiples visualizaciones, existe un esfuerzo continuo para generar extensiones de codificación de vídeo tridimensional (3DV) para H.264 y HEVC. Las extensiones 3DV proporcionan mapas de profundidad. Los mapas de profundidad son imágenes cuyos valores de píxel representan las profundidades tridimensionales de los objetos que se muestran en las correspondientes imágenes de "textura". En algunos ejemplos, los valores de píxeles más brillantes en un mapa de profundidad pueden corresponder a objetos que están más cerca de una cámara y los valores de píxeles más oscuros en un mapa de profundidad pueden corresponder a objetos que están más alejados de la cámara. Las imágenes de "textura" pueden incluir imágenes convencionales. A diferencia de 3DV, MVC no proporciona mapas de profundidad y solo las imágenes de textura están codificadas. Sin embargo, dado que las extensiones 3DV también usan múltiples visualizaciones de textura, las extensiones 3DV pueden emplear muchas de las técnicas de codificación utilizadas en MVC.

**[57]** En las extensiones 3DV para H.264/AVC y HEVC, el codificador de vídeo 20 puede codificar un mapa de profundidad de la misma manera que otras visualizaciones de una unidad de acceso. Es decir, además de imágenes de textura para diferentes visualizaciones, una unidad de acceso puede incluir un mapa de profundidad. Un mapa de profundidad para una unidad de acceso también se puede denominar un "componente de visualización de profundidad" de la unidad de acceso. El término "componente de visualización" puede usarse para referirse a los componentes de visualización de textura o al componente de visualización de profundidad de una unidad de acceso.

**[58]** Los codificadores de vídeo (por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30) pueden usar la Representación Basada en Imagen de Profundidad (DIBR) para generar, en función de los componentes de visualización de textura y profundidad disponibles, componentes de visualización de textura sintética. Un componente de visualización de textura sintética puede ser un componente de visualización de textura que se sintetiza basándose en un mapa de profundidad y uno o más componentes de visualización de textura. En algunos casos, se puede usar un componente de visualización de textura sintética como una imagen de referencia para la predicción de la unidad entre accesos o la predicción entre visualizaciones. Los componentes de visualización de textura sintética que se usan como imágenes de referencia se pueden denominar imágenes de referencia de síntesis de visualización (VSRP). Los codificadores de vídeo pueden incluir VSRP en listas de imágenes de referencia. En algunos diseños específicos, un codificador de vídeo puede usar múltiples componentes de visualización (con profundidad) para generar un VSRP a través de DIBR. En algunos ejemplos, solo un VSRP puede estar disponible para cada imagen codificada.

**[59]** Algunas imágenes de referencia son imágenes de referencia a "corto plazo" y algunas imágenes de referencia son imágenes de referencia a "largo plazo". Las imágenes de referencia a largo plazo pueden permanecer en las listas de imágenes de referencia durante períodos de tiempo más largos que las imágenes de referencia a corto plazo. Si el codificador de vídeo 20 determina que sería ventajoso mantener la disponibilidad de una imagen de referencia durante un período de tiempo prolongado (por ejemplo, varios segundos), el codificador de vídeo 20 puede señalar que la imagen de referencia es una imagen de referencia a largo plazo. Por ejemplo, el codificador de vídeo 20 puede indicar que una trama de referencia es una imagen de referencia a largo plazo si la imagen de referencia contiene un fondo estático y otras imágenes cubren y descubren partes de este fondo estático. En H.264/AVC o H.264/MVC, una imagen de referencia a corto plazo nunca tiene un número de tramas (frame\_num) igual al número de tramas de la imagen actual (es decir, la imagen que se está codificando actualmente). Un número de tramas de una imagen es un valor basado en un número de imagen de la imagen. El número de tramas de una imagen puede usarse para identificar la imagen cuando la imagen se utiliza como una imagen de referencia a corto plazo.

**[60]** Como se ha mencionado anteriormente, los codificadores de vídeo (por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30) pueden mantener una primera lista de imágenes de referencia (lista 0) y una segunda lista de imágenes de referencia (lista 1). La lista 0 y la lista 1 pueden ser listas de números de tramas de imágenes de referencia. El codificador de vídeo 20 puede señalar que un bloque de vídeo está interpuesto usando un bloque de referencia en una imagen de referencia particular señalando una posición en una lista de imágenes de referencia de los números de trama de la imagen de referencia particular.

**[61]** Algunos codificadores de vídeo generan listas de imágenes de referencia de manera que los números de tramas de las imágenes de referencia temporales (es decir, intra-visualización) siempre se enumeran inicialmente en primer lugar. Los números de trama de las imágenes de referencia temporales son seguidos en las listas de imágenes de referencia por los números de trama de las imágenes de referencia entre visualizaciones. Es decir, un codificador de vídeo puede generar una imagen de referencia usando los siguientes pasos. Primero, el codificador de vídeo puede aplicar un proceso de inicialización de la lista de imágenes de referencia para imágenes de

referencia temporales (es decir, intra-visualización) como se especifica en la norma H.264/AVC, en el que no se consideran las imágenes de referencia de otras visualizaciones. Segundo, el codificador de vídeo puede adjuntar imágenes de referencia entre visualizaciones al final de la lista de imágenes de referencia en un orden en que las imágenes de referencia entre visualizaciones se producen en la extensión de MVC SPS. En tercer lugar, el codificador de vídeo puede aplicar un proceso de modificación de lista de imágenes de referencia (RPLM) para imágenes de referencia tanto de intra-visualización como entre visualizaciones. Por ejemplo, un codificador de vídeo o un descodificador de vídeo puede realizar un proceso de modificación de la lista de imágenes de referencia que cambia una posición en una lista de imágenes de referencia de un componente de visualización de referencia entre visualizaciones. Las imágenes de referencia entre visualizaciones se pueden identificar en los comandos de RPLM por sus valores de índice, según lo especificado por la extensión de MVC SPS. El proceso RPLM se describe en detalle a continuación.

**[62]** En HEVC, una cabecera de unidad NAL de una unidad VCL NAL puede tener una longitud de dos bytes. Por el contrario, una cabecera de unidad NAL de una unidad VCL NAL en H.264/AVC solo puede tener un byte de longitud. El primer byte de la cabecera de unidad NAL de una unidad HEVC VCL NAL puede tener la misma sintaxis y semántica que la cabecera de unidad NAL de una unidad H.264/AVC VCL NAL. Cuando el segundo byte de la cabecera de unidad NAL de una unidad HEVC VCL NAL está presente, el segundo byte de la cabecera de unidad NAL de la unidad HEVC VCL NAL puede incluir un elemento sintáctico de identificador temporal (por ejemplo, "temporal\_id") y un elemento sintáctico de indicador de salida (por ejemplo, "output\_flag"). El elemento sintáctico del identificador temporal puede especificar un identificador temporal para la unidad NAL. Cada unidad NAL que incluye un fragmento codificado de un componente de visualización de una unidad de acceso puede tener el mismo identificador temporal. El elemento sintáctico del indicador de salida puede afectar a la salida de una imagen descodificada asociada con la unidad NAL, como se describe en el Anexo C del HEVC.

**[63]** Además, después de que un codificador de vídeo genere listas de imágenes de referencia 0 y 1 en HEVC, el codificador de vídeo puede generar una lista de imágenes de referencia combinada a partir de imágenes de referencia en las listas de imágenes de referencia 0 y 1. Para generar la lista de imágenes de referencia combinadas, el codificador de vídeo puede seleccionar entradas (es decir, imágenes de referencia) de las listas 0 y 1 e insertar (por ejemplo, agregar) las entradas seleccionadas a la lista de imágenes de referencia combinadas. En algunos ejemplos, el codificador de vídeo puede seleccionar las entradas de la lista 0 y la lista 1 en función del orden ascendente de ocurrencia en la lista 0 y la lista 1. Si una entrada seleccionada ya se encuentra en la lista de imágenes de referencia combinadas, el codificador de vídeo no vuelve a insertar la entrada en la lista de imágenes de referencia combinadas. El codificador de vídeo puede determinar si una entrada seleccionada ya está en la lista de imágenes de referencia combinadas comprobando los números de recuento de orden de imágenes (POC) de las entradas.

**[64]** En HEVC, el codificador de vídeo 20 señala un conjunto de imágenes de referencia (RPS) para cada imagen codificada. Un RPS de una imagen codificada es un conjunto de imágenes de referencia asociadas con la imagen codificada. El RPS puede incluir, y en algunos ejemplos consiste en, todas las imágenes de referencia que son anteriores a la imagen codificada en orden de descodificación que puede usarse para la inter-predicción de la imagen codificada o cualquier imagen que siga a la imagen codificada en orden de descodificación.

**[65]** El codificador de vídeo 20 puede señalar los RPS de imágenes codificadas en PPS y cabeceras de fragmento. Por ejemplo, el codificador de vídeo 20 puede señalar un RPS en un PPS y puede señalar en una cabecera de trama de una imagen codificada que el RPS de la imagen codificada es el RPS señalado en el PPS. En otros casos, el codificador de vídeo 20 puede señalar el RPS de una imagen codificada directamente en una cabecera de trama de la imagen codificada. En consecuencia, el descodificador de vídeo 30 puede analizar, desde una cabecera de trama, los elementos sintácticos que indican el RPS.

**[66]** El RPS de un componente de visualización (es decir, el componente de visualización actual) se puede dividir en cinco subconjuntos de imágenes de referencia: RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr, y RefPicSetLtFoll. Cada uno de estos subconjuntos de imágenes de referencia puede comprender una lista de valores POC que identifican componentes de visualización de referencia. Los términos "componentes de visualización de referencia" e "imágenes de referencia" se pueden usar indistintamente en esta divulgación. RefPicSetStCurr0 puede consistir en todos los componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual tanto en orden de descodificación como en orden de salida, y que pueden usarse en la inter-predicción del componente de visualización actual. RefPicSetStCurr1 puede consistir en todos los componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual en orden de descodificación, que vienen a continuación del componente de visualización actual en orden de salida, y que pueden usarse en inter-predicción del componente de visualización actual. RefPicSetStFoll0 puede consistir en todos los componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual tanto en orden de descodificación como en orden de salida, que pueden usarse en inter-predicción de uno o más de los componentes de visualización siguientes al componente de visualización actual en orden de descodificación, y que no se utilizan en la inter-predicción del componente de visualización actual. RefPicSetStFoll1 puede consistir en todos los componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual en orden de

descodificación, que vienen a continuación del componente de visualización actual en orden de salida, que pueden usarse en inter-predicción de uno o más de los componentes de visualización siguientes al componente de visualización actual en orden de descodificación, y que no se usan en inter-predicción del componente de visualización actual. RefPicSetLtCurr puede consistir en todos los componentes de visualización de referencia a largo plazo que son anteriores al componente de visualización actual en el orden de descodificación y que pueden usarse en inter-predicción del componente de visualización actual. RefPicSetLtFoll puede consistir en todos los componentes de visualización de referencia a largo plazo que son anteriores al componente de visualización actual en orden de descodificación, que pueden usarse en inter-predicción de uno o más componentes de visualización que vienen a continuación del componente de visualización actual en orden de descodificación, y que no se usan en la inter-predicción del componente de visualización actual.

**[67]** El número de entradas en RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr y RefPicSetLtFoll pueden denominarse en esta divulgación NumRpsStCurr0, NumRpsStCurr1, NumRpsStFoll0, NumRpsStFoll1, NumRpsLtCurr y NumRpsLtFoll respectivamente. Si el componente de visualización actual es una imagen IDR, RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr y RefPicSetLtFoll están todos vacíos, y NumRpsStCurr0, NumRpsStCurr1, NumRpsStFoll0, NumRpsStFoll1, NumRpsLtCurr y NumRpsLtFoll son todos iguales a 0. Se puede encontrar una descripción más detallada de los conjuntos de imágenes de referencia en Boyce et al., "Informe de ruptura de JCT-VC: Memoria intermedia de imágenes de referencia y construcción de lista (AHG21), documento n.º JCTVC-G1002, con fecha del 11 de noviembre de 2011, que a partir del 19 de diciembre de 2012 está disponible en [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/7\\_Geneva/wg11/JCTVC-G1002-v5.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G1002-v5.zip), el contenido completo del cual se incorpora en el presente documento como referencia.

**[68]** Cuando un codificador de vídeo codifica una trama P o B, el codificador de vídeo puede generar versiones iniciales de la lista 0 y la lista 1. En otras palabras, el codificador de vídeo puede inicializar la lista 0 y la lista 1. El codificador de vídeo puede generar las versiones iniciales de la lista 0 y la lista 1 de modo que la lista 0 y la lista 1 incluyan al menos una imagen de referencia de RefPicSetCurr0, RefPicSetCurr1 o RefPicSetLtCurr.

**[69]** Por ejemplo, el codificador de vídeo puede realizar una operación descrita por el siguiente pseudocódigo para generar la versión inicial de la lista 0.

```

cldx = 0

while(cldx <= num_ref_idx_10_active_minus1)
{
    for(i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
        RefPicList0[cldx] = RefPicSetStCurr0[i]
    for(i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
        RefPicList0[cldx] = RefPicSetStCurr1[i]
    for(i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
        RefPicList0[cldx] = RefPicSetLtCurr[i]
}

```

En el pseudocódigo anterior, cldx es una variable de contador. Además, en el pseudocódigo anterior, num\_ref\_idx\_10\_active\_minus1 es un valor que indica el número de imágenes de referencia activas en la lista 0, menos 1. Cuando el codificador de vídeo ejecuta el pseudocódigo anterior, el codificador de vídeo inserta cada imagen de referencia de RefPicSetStCurr0 en la lista de imágenes de referencia 0 y luego inserta cada imagen de referencia de RefPicSetStCurr1 en la lista de imágenes de referencia 0 y luego inserta cada imagen de referencia de RefPicSetLtCurr en la lista de imágenes de referencia 0.

**[70]** El codificador de vídeo puede realizar una operación descrita por el siguiente pseudocódigo para generar la versión inicial de la lista 1.

```

cldx = 0

while(cldx <= num_ref_idx_11_active_minus1)
{

```

```

for(i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
    RefPicList1[cldx] = RefPicSetStCurr1[i]
5
for(i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
    RefPicList1[cldx] = RefPicSetStCurr0[i]
10
for(i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
    RefPicList1[cldx] = RefPicSetLtCurr[i]
15
}

```

Cuando el codificador de vídeo ejecuta el pseudocódigo anterior, el codificador de vídeo inserta cada imagen de referencia de RefPicSetStCurr0 en la lista de imágenes de referencia 1 y luego inserta cada imagen de referencia de RefPicSetStCurr1 en la lista de imágenes de referencia 1 y luego inserta cada imagen de referencia de RefPicSetLtCurr en la lista de imágenes de referencia 1.

[71] En algunos casos, puede ser ventajoso cambiar el orden de las imágenes de referencia en una lista de imágenes de referencia. Por ejemplo, debido a que las posiciones pueden señalarse usando valores unarios, pueden requerirse menos bits para indicar la primera posición en una lista de imágenes de referencia que una última posición en la lista de imágenes de referencia. Por lo tanto, si es probable que una imagen de referencia particular se use con frecuencia, puede ser ventajoso tener la imagen de referencia particular más cerca de la primera posición de la lista de imágenes de referencia que de la última posición en la lista de imágenes de referencia.

[72] En consecuencia, el codificador de vídeo 20 puede incluir una serie de uno o más comandos RPLM en el flujo de bits. Un comando RPLM puede ser un conjunto de uno o más elementos sintácticos para insertar una imagen de referencia en una lista de imágenes de referencia. De este modo, al incluir comandos de RPLM en el flujo de bits, el codificador de vídeo 20 puede disponer de forma flexible referencias de predicción temporales y de visualización, que pueden proporcionar posibles ganancias de eficiencia de codificación. Además, el uso de comandos RPLM puede aumentar la capacidad de recuperación ante errores porque la selección de imágenes de referencia y los mecanismos de imágenes redundantes pueden extenderse a la dimensión de visualización.

[73] El descodificador de vídeo 30 procesa los comandos RPLM en el orden en que los comandos RPLM se señalan en el flujo de bits. Además, cuando el descodificador de vídeo 30 procesa una serie de comandos de RPLM, el descodificador de vídeo 30 puede establecer inicialmente un valor de índice actual a 0 y puede incrementar el valor de índice actual durante el procesamiento de cada comando de RPLM. Cuando el descodificador de vídeo 30 procesa un comando RPLM, el descodificador de vídeo 30 puede desplazar hacia abajo en la lista de imágenes de referencia, una posición, la imagen de referencia en la posición indicada por el valor de índice actual y todas las imágenes de referencia en posiciones siguientes a la posición indicada por el valor de índice actual. El descodificador de vídeo 30 puede entonces insertar una imagen de referencia en la lista de imágenes de referencia en la posición indicada por el valor de índice actual. El descodificador de vídeo 30 puede entonces escanear a través de la lista de imágenes de referencia y eliminar, si está presente, un duplicado de la imagen de referencia insertada.

[74] Los comandos de RPLM se pueden especificar en una cabecera de fragmento de un fragmento codificado. La Tabla 3 siguiente muestra una sintaxis de ejemplo para los comandos RPLM que pueden incluirse en una cabecera de trama.

**TABLA 3 - Sintaxis RPLM**

|  | Descriptor   |
|--|--------------|
| ref_pic_list_modification(){             |              |
| if(slice_type!= 2){                      |              |
| <b>ref_pic_list_modification_flag_I0</b> | u(1)         |
| if(ref_pic_list_modification_flag_I0)    |              |
| do {                                     |              |
| <b>list_modification_idc</b>             | <u>ue(v)</u> |
| if(list_modification_idc != 3)           |              |
| <b>ref_pic_set_idx</b>                   | <u>ue(v)</u> |
| } while(list_modification_idc != 3)      |              |
| }  |              |
| if(slice_type % 5 == 1){                 |              |
| <b>ref_pic_list_modification_flag_I1</b> | u(1)         |
| if(ref_pic_list_modification_flag_I1)    |              |

|                                     |              |
|-------------------------------------|--------------|
| do {                                |              |
| <b>list_modification_idc</b>        | <u>ue(v)</u> |
| if(list_modification_idc != 3)      |              |
| <b>ref_pic_set_idx</b>              | <u>ue(v)</u> |
| } while(list_modification_idc != 3) |              |
| }                                   |              |
| }                                   |              |

[75] En la Tabla 3, "slice\_type" indica un tipo del fragmento actual (es decir, el fragmento cuya cabecera de fragmento incluye los comandos RPLM). La Tabla 4 siguiente indica la semántica de ejemplo de "slice\_type".

5

**TABLA 4**

| slice_type | Nombre de slice_type |
|------------|----------------------|
| 0          | P (fragmento P)      |
| 1          | B (fragmento B)      |
| 2          | I (fragmento I)      |

[76] Además, en la sintaxis de ejemplo de la Tabla 3, los elementos sintácticos "list modification idc" y "ref\_pic\_set\_idx" pueden especificar cambios de las versiones iniciales de las listas de imágenes de referencia a las versiones de las listas de imágenes de referencia que un codificador de vídeo usa para la inter-predicción. El elemento sintáctico "ref\_pic\_list\_modification\_flag\_I0" indica si hay uno o más elementos sintácticos "list\_modification\_idc" para modificar la lista 0. Cuando el elemento sintáctico "ref\_pic\_list\_modification\_flag\_I0" es igual a 1, el número de veces que el elemento sintáctico "list\_modification\_idc" no es igual a 3 puede no exceder el número de listas de imágenes de referencia activas en la lista 0. El elemento sintáctico "ref\_pic\_list\_modification\_flag\_I1" indica si hay uno o más elementos sintácticos "list\_modification\_idc" para modificar la lista 1. Cuando "ref\_pic\_list\_modification\_flag\_I1" es igual a 1, el número de veces que list\_modification\_idc no es igual a 3 puede no exceder el número de listas de imágenes de referencia activas en la lista 1. El elemento sintáctico "long\_term\_pic\_num" especifica un número de tramas a largo plazo de una imagen que se mueve al índice actual de la lista de imágenes de referencia.

[77] En general, el elemento sintáctico "ref\_pic\_set\_idx" especifica un índice para una posición en RefPicSetStCurr0, RefPicSetStCurr1 o RefPicSetLtCurr de una imagen de referencia que se moverá al índice actual en la lista de imágenes de referencia. El valor del elemento sintáctico "ref\_pic\_set\_idx" puede variar desde 0 hasta el número máximo permitido de imágenes de referencia, inclusive.

[78] El elemento sintáctico "list\_modification\_idc" especifica un tipo de comando RPLM para realizar. Para facilitar la explicación, este elemento sintáctico puede denominarse elemento sintáctico de tipo RPLM. En algunos ejemplos, el elemento sintáctico de tipo RPLM puede tener un valor que varía de 0 a 3, inclusive. Para la lista 0, si el elemento sintáctico de tipo RPLM es igual a 0, el elemento sintáctico "ref\_pic\_set\_idx" está presente y el elemento sintáctico "ref\_pic\_set\_idx" corresponde a un índice de una posición en RefPicSetStCurr0. Por ejemplo, "ref\_pic\_set\_idx" puede indicar la imagen de referencia en la segunda posición en RefPicSetStCurr0. Para la lista 1, si el elemento sintáctico de tipo RPLM es igual a 0, el elemento sintáctico "ref\_pic\_set\_idx" está presente y el elemento sintáctico "ref\_pic\_set\_idx" corresponde a un índice de una posición en RefPicSetStCurr1.

[79] Para la lista 0, si el elemento sintáctico de tipo RPLM es igual a 1, el elemento sintáctico "ref\_pic\_set\_idx" está presente y el elemento sintáctico "ref\_pic\_set\_idx" corresponde a un índice de una posición en RefPicSetStCurr1. Para la lista 1, si el elemento sintáctico de tipo RPLM es igual a 1, el elemento sintáctico "ref\_pic\_set\_idx" está presente y el elemento sintáctico "ref\_pic\_set\_idx" corresponde a un índice de una posición en RefPicSetStCurr0.

[80] Si el elemento sintáctico de tipo RPLM es igual a 2, el elemento sintáctico "ref\_pic\_set\_idx" está presente y el elemento sintáctico "ref\_pic\_set\_idx" indica un índice a una posición en RefPicSetLtCurr. Si el elemento sintáctico de tipo RPLM es igual a 3, el codificador de vídeo detiene el ciclo de modificación de la lista de imágenes de referencia.

[81] Cuando el elemento sintáctico "ref\_pic\_list\_modification\_flag\_I0" es igual a 1, un codificador de vídeo puede procesar los elementos sintácticos "list\_modification\_idc" en el orden en que ocurren en el flujo de bits. Para cada elemento sintáctico "list\_modification\_idc", el codificador de vídeo puede determinar si el elemento sintáctico "list\_modification\_idc" es igual a 0, 1 o 2. Si el elemento sintáctico "list\_modification\_idc" es igual a 0, 1 o 2, el codificador de vídeo puede invocar un proceso para mover una imagen de referencia a un índice de imagen de referencia. El codificador de vídeo puede proporcionar un índice refIdxL0 como entrada al proceso y puede establecer refIdxL0 a la salida del proceso. RefIdxL0 es un índice en la lista 0. El codificador de vídeo inicialmente puede establecer que refIdxL0 sea igual a 0. El proceso para mover una imagen de referencia a un índice de imagen de referencia se describe a continuación. De lo contrario, si el elemento sintáctico "list\_modification\_idc" es igual a 3,

el codificador de vídeo detiene el proceso de modificación de la lista de imágenes de referencia para la lista 0.

**[82]** Cuando un fragmento actual es un fragmento B y el elemento sintáctico "ref\_pic\_list\_modification\_flag\_l1" es igual a 1, el codificador de vídeo puede procesar los elementos sintácticos "list\_modification\_idc" en el orden en que ocurren en el flujo de bits. Para cada elemento sintáctico "list\_modification\_idc", el codificador de vídeo puede determinar si el elemento sintáctico "list\_modification\_idc" es igual a 0, 1 o 2. Si el elemento sintáctico "list\_modification\_idc" es igual a 0, 1 o 2, el codificador de vídeo puede invocar un proceso para mover una imagen de referencia a un índice de imagen de referencia. El codificador de vídeo puede proporcionar un índice reldxL1 como entrada al proceso y puede establecer reldxL1 a la salida del proceso. RefldxL1 es un índice en la lista 1. El codificador de vídeo inicialmente puede establecer que reldxL1 sea igual a 0. El proceso para mover una imagen de referencia a un índice de imagen de referencia se describe a continuación. De lo contrario, si el elemento sintáctico "list\_modification\_idc" es igual a 3, el codificador de vídeo detiene el proceso de modificación de la lista de imágenes de referencia para la lista 1.

**[83]** Como se ha mencionado anteriormente, un codificador de vídeo puede realizar un proceso para mover una imagen de referencia a un índice de imagen de referencia. Este proceso puede tomar el índice de imagen de referencia como entrada y generar un índice de imagen de referencia incrementado. El índice de imagen de referencia se puede denotar como reldxLX, donde X es igual a 0 para el movimiento de una imagen de referencia en la lista 0 y X es igual a 1 para el movimiento de una imagen de referencia en la lista 1. Si el elemento sintáctico "list\_modification\_idc" es igual a 0 y la lista de imágenes de referencia actual es la lista 0, el descodificador de vídeo puede designar RefPicSetStCurr0 como el conjunto de imágenes de referencia actual. De lo contrario, si el elemento sintáctico "list\_modification\_idc" es igual a 0 y la lista de imágenes de referencia actual es la lista 1, el codificador de vídeo puede designar RefPicStCurr1 como el conjunto de imágenes de referencia actual. Si el elemento sintáctico "list\_modification\_idc" es igual a 1 y la lista de imágenes de referencia actual es la lista 0, el codificador de vídeo puede designar RefPicStCurr1 como el conjunto de imágenes de referencia actual. Si el elemento sintáctico "list\_modification\_idc" es igual a 1 y la lista de imágenes de referencia actual es la lista 1, el codificador de vídeo puede designar RefPicStCurr0 como el conjunto de imágenes de referencia actual. Si el elemento sintáctico "list\_modification\_idc" es igual a 2, el codificador de vídeo puede designar RefPicSetLtCurr como el conjunto de imágenes de referencia actual.

**[84]** Además, en el proceso para mover una imagen de referencia a un índice de imagen de referencia, el codificador de vídeo puede determinar un valor POC relevante, pocLX. El valor POC relevante puede ser el valor POC de una imagen de referencia para insertar en la lista de imágenes de referencia X (donde X es 0 o 1) en la posición en el conjunto de imágenes de referencia actual indicada por el elemento sintáctico "ref\_pic\_set\_idx". Además, el codificador de vídeo puede realizar un proceso descrito por el siguiente pseudocódigo para insertar una imagen que tenga el valor POC relevante en la lista de imágenes de referencia actual en la posición indicada por reldxLX.

```

for(cldx = num_ref_idx_1X_active_minus1 + 1; cldx > reldxLX; cldx--)
    RefPicListX[cldx] = RefPicListX[cldx - 1]
    RefPicListX[reldxLX++] = pocLX
nldx = reldxLX
for(cldx = reldxLX; cldx <= num_ref_idx_1X_active_minus1 + 1; cldx++)
    if(PicOrderCnt(RefPicListX[cldx])!= pocLX)
        RefPicListX[nldx++] = RefPicListX[cldx]

```

En el pseudocódigo anterior, cldx es una variable de contador. Además, en el pseudocódigo anterior, num\_ref\_idx\_1X\_active\_minus1 es una variable igual al número de imágenes de referencia activas en la lista de imágenes de referencia actual, menos 1. PicOrderCnt es una función que devuelve un valor POC de una imagen de referencia. Cuando el codificador de vídeo realiza el proceso descrito por el pseudocódigo anterior, el codificador de vídeo puede desplazar las imágenes de referencia siguiendo la posición indicada por reldxLX a las posiciones posteriores en la lista de imágenes de referencia actual, insertar la imagen de referencia con el valor POC relevante, incrementar reldxLX, y eliminar de la lista de imágenes de referencia cualquier duplicado de la imagen de referencia con el valor POC relevante.

**[85]** En el pseudocódigo anterior, la longitud de la lista de imágenes de referencia actual puede ser temporalmente un elemento más largo que la longitud de la versión final de la lista de imágenes de referencia actual. En consecuencia, después de que el codificador de vídeo realice la operación descrita por el pseudocódigo anterior, el codificador de vídeo puede retener solo los elementos 0 a num\_ref\_idx\_1X\_active\_minus\_1 en la versión final de la lista de imágenes de referencia actual.

5 [86] La Solicitud de Patente Provisional de Estados Unidos N.º 61/512 771, cuyo contenido completo se incorpora en el presente documento como referencia, describe una extensión de HEVC para incluir la codificación de múltiples visualizaciones de textura, similar a la extensión de MVC de H.264/AVC. Es decir, la Solicitud de Patente Provisional de Estados Unidos N.º 61/512 771, describe el uso en HEVC de la predicción entre visualizaciones (es decir, compensación de movimiento de disparidad), similar a la utilizada en H.264/AVC. Además, en la Solicitud de Patente Provisional de Estados Unidos N.º 61/512 771, un codificador de vídeo puede señalar, en un SPS, los identificadores de visualización de los componentes de visualización de los que dependen las visualizaciones que no son de base. La Tabla 5, a continuación, es una parte de una sintaxis de un SPS para señalar las dependencias de visualización de las visualizaciones que no son de base.

TABLA 5

|   |       |
|---|-------|
| for(i = 1; i <= imm_views_mmus1; i++) { |       |
| num_ref_views[i]                        | ue(v) |
| for(j = 0; j < num_ref_views[i]; j++)   |       |
| ref_view_idx[i][j]                      | ue(v) |
| }                                       |       |

15 [87] En la Tabla 5 anterior, los elementos sintácticos "ref\_view\_idx[i][j]" indican los índices de orden de visualización del componente de visualización j-ésimo para la predicción entre visualizaciones en las versiones iniciales de la lista de imágenes de referencia 0 y la lista de imágenes de referencia 1 para descodificación un componente de visualización con índice de orden de visualización igual a i. En la Solicitud de Patente Provisional de los Estados Unidos N.º 61/512 771, la construcción de la lista de imágenes de referencia sigue el mismo procedimiento que en H.264/MVC. Sin embargo, en la Solicitud de Patente Provisional de Estados Unidos N.º 61/512 771, el codificador de vídeo 20 puede usar la misma indicación, independientemente de si la imagen actual es una imagen de anclaje o sin anclaje y si la lista actual es RefPicList0 o RefPicList1. De esta manera, se pueden usar dos indicaciones para los comandos RPLM y la diferencia del índice de referencia entre visualizaciones se puede usar para identificar una imagen de referencia entre visualizaciones.

25 [88] Puede haber varios problemas con los codificadores de vídeo descritos anteriormente. Por ejemplo, la indicación de los componentes de visualización de referencia entre visualizaciones puede ser ineficaz o inexistente. En otro ejemplo, el proceso de inicialización de la lista de imágenes de referencia puede ser ineficiente. En otro ejemplo, el proceso de modificación de la lista de imágenes de referencia puede ser ineficiente.

30 [89] Las técnicas de esta divulgación pueden remediar estos problemas. De acuerdo con las técnicas de esta divulgación, un codificador de vídeo obtiene un RPS para un componente de visualización actual. El RPS del componente de visualización actual puede incluir los subconjuntos de imágenes de referencia: RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, Ref-PicSetLtCurr, RefPicSetLtFoll. Además, el RPS del componente de visualización actual puede incluir un conjunto de imágenes de referencia entre visualizaciones (RefPicSetInterView). El conjunto de imágenes de referencia de predicción entre visualizaciones incluye componentes de visualización que no son de una misma visualización que el componente de visualización actual, que también son de una misma unidad de acceso que el componente de visualización actual, y que también son utilizados por el componente de visualización actual para referencia de predicción entre visualizaciones. En algunos ejemplos, el conjunto de imágenes de referencia entre visualizaciones se señala a nivel de secuencia y puede ser compartido por todos los componentes de visualización de la misma visualización. RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr, y RefPicSetLtFoll incluyen componentes de visualización de referencia que son de la misma visualización que el componente de visualización actual, son anteriores al componente de visualización actual en orden de descodificación y se pueden usar para la interpredicción del componente de visualización actual o cualquier componente de visualización de la misma visualización siguiente al componente de visualización actual en orden de descodificación. RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr, y RefPicSetLtFoll se pueden denominar en el presente documento el conjunto de imágenes de referencia entre visualizaciones.

50 [90] En un ejemplo, el codificador de vídeo 20 puede obtener RefPicSetInterView utilizando el siguiente pseudocódigo:

```

for(j = 0; j < num_ref_views[VOIdx]; j++)
    RefPicSetInterView[j] = ref_view_idx[VOIdx][j]
    NumRpsInterView = j
    
```

60 En el pseudocódigo anterior, VOIdx indica el índice de orden de visualización del componente de visualización actual.

**[91]** En otro ejemplo, el codificador de vídeo 20 puede obtener RefPicSetInterView utilizando el siguiente pseudocódigo:

```

5      if(all_inter_view_refs_flag){
          for(j = 0; j < num_ref_views[VOIdx]; j++)
              RefPicSetInterView[j] = ref_view_idx[VOIdx][j]
10         NumRpsInterView = j
        }
    else if(no_inter_view_refs_flag)
15         NumRpsInterView = 0 // RefPicSetInterView is empty
    else {
20         for(j = 0, jj = 0; jj < num_ref_views[VOIdx]; jj++)
            if(ref_view_flag[VOIdx][jj])
25                 RefPicSetInterView[jj] = ref_view_idx[VOIdx][jj]
            NumRpsInterView = j
        }

```

30 En el pseudocódigo anterior, los valores de ref\_view\_flag[VOIdx][jj] para todos los valores de j provienen de la estructura sintáctica inter\_view\_rps(i, inter\_view\_rps\_idx) en el conjunto de parámetros de secuencia activa.

**[92]** Por ejemplo, el codificador de vídeo 20 puede generar, basándose en un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso, una lista de imágenes de referencia para el componente de visualización actual. El conjunto de imágenes de referencia puede incluir un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones. Además, el codificador de vídeo 20 puede codificar el componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia. El codificador de vídeo 20 puede generar un flujo de bits que incluye elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual.

**[93]** Además, de acuerdo con las técnicas de esta divulgación, el decodificador de vídeo 30 puede analizar, a partir de un flujo de bits, elementos sintácticos que indican un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso, incluyendo el conjunto de imágenes de referencia un conjunto de imágenes de referencia entre visualizaciones. que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones. El decodificador de vídeo 30 puede generar, basándose en el conjunto de imágenes de referencia, una lista de imágenes de referencia para el componente de visualización actual. Además, el decodificador de vídeo 30 puede decodificar el componente de visualización actual basándose al menos en parte en una o más imágenes de referencia en la lista de imágenes de referencia.

**[94]** Cuando un codificador de vídeo (como el codificador de vídeo 20 o el decodificador de vídeo 30) inicializa las listas de imágenes de referencia 0 y 1 para el componente de visualización actual, el codificador de vídeo incluye componentes de visualización de referencia de RefPicSetInterView en las listas de imágenes de referencia 0 y 1. En algunos ejemplos, el codificador de vídeo genera listas de imágenes de referencia 0 y 1 de manera que las listas de imágenes de referencia 0 y 1 incluyen cada componente de visualización del conjunto de imágenes de referencia entre visualizaciones (RefPicSetInterView). El codificador de vídeo puede incluir componentes de visualización de referencia de RefPicSetInterView en varias ubicaciones en las listas de imágenes de referencia 0 y 1 de acuerdo con varias órdenes predefinidas.

**[95]** En un primer orden de ejemplo, el codificador de vídeo inicializa la lista de imágenes de referencia 0 de modo que los componentes de visualización en RefPicSetStCurr0 se enumeran primero, seguidos de los componentes de visualización en RefPicSetInterView, los componentes de visualización en RefPicSetStCurr1 y los componentes de visualización en RefPicSetLtCurr. En este ejemplo, el codificador de vídeo inicializa la lista de imágenes de referencia 1 de forma tal que los componentes de visualización en RefPicSetStCurr1 se enumeran primero, seguidos de los componentes de visualización en RefPicSetInterView, los componentes de visualización en

RefPicSetStCurr0 y los componentes de visualización en RefPicSetLtCurr. Así, en este ejemplo, el descodificador de vídeo 30 puede generar una lista de imágenes de referencia (por ejemplo, lista 0 o lista 1) de manera que un primer subconjunto (es decir, RefPicSetStCurr0) o un segundo subconjunto (es decir, RefPicSetStCurr1) se encuentre primero en la lista de imágenes de referencia, seguido del conjunto de imágenes de referencia entre visualizaciones (es decir, RefPicSetInterView).

**[96]** En un segundo orden de ejemplo, el codificador de vídeo inicializa la lista de imágenes de referencia 0 de modo que los componentes de visualización en RefPicSetStCurr0 se enumeran primero, seguidos de los componentes de visualización en RefPicSetStCurr1, los componentes de visualización en RefPicSet-InterView y los componentes de visualización en RefPicSetLtCurr. En este ejemplo, el codificador de vídeo inicializa la lista de imágenes de referencia 1 de modo que los componentes de visualización en RefPicSetStCurr1 se enumeran primero, seguidos de los componentes de visualización en RefPicSetStCurr0, los componentes de visualización en RefPicSetInterView y los componentes de visualización en RefPicSetLtCurr. Por lo tanto, en este ejemplo, el descodificador de vídeo 30 puede generar una lista de imágenes de referencia (por ejemplo, lista 0 o lista 1) de manera que un primer subconjunto (es decir, RefPicSetStCurr0) y un segundo subconjunto (es decir, RefPicSetStCurr1) ocurran en la lista de imágenes de referencia, seguidos del conjunto de imágenes de referencia entre visualizaciones (es decir, RefPicSetInterView).

**[97]** En un tercer orden de ejemplo, el codificador de vídeo inicializa la lista de imágenes de referencia 0 de modo que los componentes de visualización en RefPicSetStCurr0 se enumeran primero, seguidos de los componentes de visualización en RefPicSetStCurr1, los componentes de visualización en RefPicSetLtCurr y los componentes de visualización en RefPicSetInterView. En este ejemplo, el codificador de vídeo inicializa la lista de imágenes de referencia 1 de modo que los componentes de visualización en RefPicSetStCurr1 se enumeran primero, seguidos de los componentes de visualización en RefPicSetStCurr0, los componentes de visualización en RefPicSetLtCurr y los componentes de visualización en RefPicSetInterView. Por lo tanto, en este tercer ejemplo, los componentes de visualización del RefPicSetInterView (el conjunto de imágenes de referencia entre visualizaciones) aparecen en último lugar en las listas de imágenes de referencia 0 y 1. Es decir, en este ejemplo, el descodificador de vídeo 30 puede generar una lista de imágenes de referencia (por ejemplo, lista 0 o lista 1) tal que un primer subconjunto (es decir, RefPicSetStCurr0), un segundo subconjunto (es decir, RefPicSetStCurr1), un quinto subconjunto (es decir, RefPicSetLtCurr) aparecen en la lista de imágenes de referencia, seguidas del conjunto de imágenes de referencia entre visualizaciones (es decir, RefPicSetInterView).

**[98]** En algunos ejemplos, RefPicSetInterView se subdivide en dos subconjuntos: RefPicSetInterViewFwd y RefPic-SetInterViewBwd. RefPicSetInterViewFwd incluye los componentes de visualización de referencia de RefPicSetInterView que tienen identificadores de visualización menores que un identificador de visualización del componente de visualización actual. RefPicSetInterViewBwd incluye los componentes de visualización de referencia de RefPicSetInterView que tienen identificadores de visualización mayores que el identificador de visualización del componente de visualización actual. El orden de los componentes de visualización en RefPicSetInterViewFwd y el orden de las visualizaciones en RefPicSetInter-ViewBwd son los mismos que los que se señalan en un SPS correspondiente. Es decir, para cualquier visualización A y B, si las visualizaciones A y B tienen valores view\_id mayores que view\_id de la visualización actual y la visualización A precede a la visualización B en el SPS, entonces un componente de visualización de la visualización A precede a un componente de visualización de la visualización B en RefPicSetInterViewBwd. Si las visualizaciones A y B tienen valores de view\_id menores que el view\_id de la visualización actual y la visualización A es anterior a la visualización B en SPS, entonces el componente de visualización de la visualización A precede al componente de visualización de la visualización B en RefPicSetInterViewFwd.

**[99]** En otros ejemplos, RefPicSetInterViewFwd incluye aquellos componentes de visualización de referencia de RefPicSetInterView que tienen índices de orden de visualización menores que un índice de orden de visualización del componente de visualización actual. RefPicSetInterViewBwd incluye los componentes de visualización de referencia de RefPicSetInterView que tienen índices de orden de visualización mayores que el índice de orden de visualización del componente de visualización actual. El orden de los componentes de visualización en RefPicSetInterViewFwd y el orden de las visualizaciones en RefPicSetInterViewBwd son los mismos que se señalan en un SPS correspondiente. Es decir, para cualquier visualización A y visualización B, si las visualizaciones A y B tienen valores de índice de orden de visualización mayores que el índice de orden de visualización de la visualización actual y la visualización A precede a la visualización B en el SPS, entonces un componente de visualización de la visualización A precede a un componente de visualización de la visualización B en RefPicSetInterViewBwd. Si las visualizaciones A y B tienen valores de índice de orden de visualización menores que el índice de orden de visualización de la visualización actual y la visualización A precede a la visualización B en el SPS, entonces el componente de visualización de la visualización A precede al componente de visualización de la visualización B en RefPicSetInterViewFwd.

**[100]** En algunos ejemplos, RefPicSetInterView se obtiene usando elementos sintácticos de dependencia de visualización como se señala en la extensión de MVC del conjunto de parámetros de secuencia de MVC. En dichos ejemplos, RefPicSetInterViewFwd puede obtenerse para que sea igual a anchor\_ref\_10[i][j] cuando el componente de visualización actual pertenece a una unidad de acceso de anclaje, que es equivalente a una unidad de acceso de

acceso aleatorio limpio (CRA). RefPicSetInterViewFwd se puede obtener para que sea igual a non\_anchor\_ref\_10[i][j] cuando el componente de visualización actual pertenece a una unidad de acceso sin anclaje, para todos los valores de j. RefPicSetInterViewBwd se obtiene para que sea igual a anchor\_ref\_11[i][j] cuando el componente de visualización actual pertenece a una unidad de acceso de anclaje, que es equivalente a una unidad de acceso CRA. RefPicSetInterViewBwd puede obtenerse para que sea igual a non\_anchor\_ref\_10[i][j] cuando el componente de visualización actual pertenece a una unidad de acceso sin anclaje, para todos los valores j.

**[101]** En algunos ejemplos donde RefPicSetInterView se subdivide en RefPicSetInterViewFwd y RefPicSetInterViewBwd, el codificador de vídeo puede inicializar la lista de imágenes de referencia 0 de manera que la lista de imágenes de referencia 0 incluya componentes de visualización de referencia de RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetLtCurr y RefPicSetInterViewFwd, pero no componentes de visualización de referencia de RefPicSetInterViewBwd. El codificador de vídeo puede inicializar la lista de imágenes de referencia 1 de modo que la lista de imágenes de referencia 1 incluya componentes de visualización de referencia de RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetLtCurr y RefPicSetInterViewBwd, pero no los componentes de visualización de referencia de RefPicSetInterViewFwd.

**[102]** Por lo tanto, el codificador de vídeo puede generar una primera lista de imágenes de referencia (por ejemplo, lista 0) de manera que la primera lista de imágenes de referencia incluye un primer subconjunto de las imágenes de referencia entre visualizaciones (por ejemplo, RefPicInterViewFwd) y no un segundo subconjunto de las imágenes de referencia entre visualizaciones (por ejemplo, RefPicInterViewBwd). Además, el codificador de vídeo puede generar una segunda lista de imágenes de referencia (por ejemplo, lista 1) de modo que la segunda lista de imágenes de referencia incluye el segundo subconjunto de las imágenes de referencia entre visualizaciones (por ejemplo, RefPicInterViewBwd) y no el primer subconjunto de las imágenes de referencia entre visualizaciones (por ejemplo, RefPicInterViewFwd).

**[103]** De forma alternativa, en algunos ejemplos donde RefPicSetInterView se subdivide en RefPicSetInterViewFwd y RefPicSetInterViewBwd, el codificador de vídeo puede inicializar las listas de imágenes de referencia 0 y 1 de acuerdo con el primer, segundo o tercer ejemplo de órdenes descritas anteriormente. Sin embargo, cuando el codificador de vídeo incluye componentes de visualización de referencia entre visualizaciones en la lista de imágenes de referencia 0, el codificador de vídeo puede incluir cada componente de visualización de referencia entre visualizaciones de RefPicSetInterViewFwd antes de los componentes de visualización de referencia entre visualizaciones de RefPicSetInterViewBwd. Cuando el codificador de vídeo incluye componentes de visualización de referencia entre visualizaciones en la lista de imágenes de referencia 1, el codificador de vídeo puede incluir cada componente de visualización de referencia entre visualizaciones de RefPicSetInterViewBwd antes de cualquier componente de visualización de referencia entre visualizaciones de RefPicSetInterViewFwd.

**[104]** En algunos ejemplos, un codificador de vídeo puede generar listas de imágenes de referencia 0 y 1 de manera que las listas de imágenes de referencia incluyen un conjunto de uno o más VSRP. El codificador de vídeo puede incluir los VSRP en varias posiciones dentro de las listas de imágenes de referencia 0 y 1. Por ejemplo, el codificador de vídeo puede inicializar las listas de imágenes de referencia 0 y 1 de manera que los VSRP se encuentren inmediatamente después de los componentes de visualización de referencia entre visualizaciones. En otras palabras, el codificador de vídeo puede generar listas de imágenes de referencia 0 y 1 de manera que el conjunto de VSRP sigue, en las listas de imágenes de referencia 0 y/o 1, al conjunto de imágenes de referencia entre visualizaciones. En otro ejemplo, el codificador de vídeo puede inicializar las listas de imágenes de referencia 0 y 1 de manera que los VSRP estén inmediatamente antes (es decir, precedan) que el conjunto de imágenes de referencia entre visualizaciones.

**[105]** En el contexto de la codificación de vídeo 3D, donde un componente de visualización puede contener tanto componentes de visualización de textura como de profundidad, un componente de visualización de textura o profundidad de una visualización específica de una unidad de acceso específica puede identificarse además con una indicación. En un ejemplo, esta indicación es un indicador en una cabecera de unidad NAL. En otro ejemplo, esta indicación se basa en tipos de unidades NAL. Las técnicas de esta divulgación, incluido el uso de un conjunto de imágenes de referencia entre visualizaciones, las técnicas de construcción de conjuntos de imágenes de referencia de esta divulgación, las técnicas de inicialización de listas de imágenes de referencia de esta divulgación, y las técnicas de modificación de listas de imágenes de referencia de esta divulgación, pueden aplicarse tanto a los componentes de visualización de textura como a los componentes de visualización de profundidad; la FIG. 4 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que está configurado para implementar las técnicas de esta divulgación. La FIG. 4 se proporciona con fines de explicación y no debe considerarse limitante de las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta divulgación describe el codificador de vídeo 20 en el contexto de la codificación HEVC, y las extensiones de MVC y 3DV de HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

**[106]** En el ejemplo de la FIG. 4, el codificador de vídeo 20 incluyen una unidad de procesamiento de predicción 100, una unidad de generación residual 102, una unidad de procesamiento de transformación 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de procesamiento de transformación

inversa 110, una unidad de reconstrucción 112, una unidad de filtro 113, una memoria intermedia de imágenes descodificadas 114, y una unidad de codificación por entropía 116. La unidad de procesamiento de predicción 100 incluye una unidad de procesamiento de inter-predicción 121 y una unidad de procesamiento de intra-predicción 126. La unidad de procesamiento de inter-predicción 121 incluye una unidad de estimación de movimiento 122 y una  
 5 unidad de compensación de movimiento 124. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales.

**[107]** El codificador de vídeo 20 recibe datos de vídeo. Para codificar los datos de vídeo, el codificador de vídeo 20 puede codificar cada fragmento de cada imagen (por ejemplo, componente de visualización de textura o  
 10 componente de visualización de profundidad) de los datos de vídeo. Como parte de codificación en un fragmento, el codificador de vídeo 20 puede codificar cada CTB en el fragmento. Como parte de la codificación de un CTB, la unidad de procesamiento de predicción 100 puede realizar la división de árbol cuádruple en el bloque de píxeles asociado con el CTB para dividir el bloque de píxeles en bloques de píxeles progresivamente más pequeños. Por  
 15 ejemplo, la unidad de procesamiento de predicción 100 puede dividir un bloque de píxeles de un CTB en cuatro sub-bloques de igual tamaño, dividir uno o más de los sub-bloques en cuatro sub-sub-bloques de igual tamaño, etc. Los bloques de píxeles más pequeños se pueden asociar con CU del CTB.

**[108]** El codificador de vídeo 20 puede codificar CU de un CTB para generar representaciones codificadas de las CU (es decir, CU codificadas). Como parte de la codificación de una CU, la unidad de procesamiento de predicción  
 20 100 puede dividir el bloque de píxeles de la CU entre una o más PU de la CU. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden soportar varios tamaños de PU. Suponiendo que el tamaño de una CU particular sea  $2N \times 2N$ , el codificador de vídeo 20 y el descodificador de vídeo 30 pueden soportar tamaños de PU de  $2N \times 2N$  o  $N \times N$  para intra-predicción, y tamaños de PU simétricos de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$  o similares para inter-predicción. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden soportar división asimétrica  
 25 para tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$  para inter-predicción.

**[109]** La unidad de procesamiento de inter-predicción 121 puede generar listas de imágenes de referencia para fragmentos de componentes de visualización. Además, la unidad de procesamiento de inter-predicción 121 puede  
 30 generar datos predictivos para una PU realizando una inter-predicción en cada PU de una CU. Los datos predictivos para la PU pueden incluir un bloque de píxeles predictivo que corresponde a la PU y la información de movimiento para la PU. Los fragmentos pueden ser fragmentos I, fragmentos P o fragmentos B. La unidad de inter-predicción 121 puede realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento en I, todas las PU son intra-predichas. Por lo tanto, si la PU está  
 35 en un fragmento I, la unidad de inter-predicción 121 no realiza inter-predicción ni predicción entre visualizaciones en la PU.

**[110]** Si una PU está en una trama P, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia en una lista de imágenes de referencia (por ejemplo, "lista 0") para un bloque de referencia para la PU. La lista 0 puede incluir una o más imágenes de referencia entre visualizaciones. El bloque de referencia de la PU puede  
 40 ser un bloque de píxeles que se corresponda más aproximadamente al bloque de píxeles de la PU. La unidad de estimación de movimiento 122 puede generar un índice de imagen de referencia que indica la imagen de referencia en la lista 0 que contiene el bloque de referencia de la PU y un vector de movimiento que indica un desplazamiento espacial entre el bloque de píxeles de la PU y el bloque de referencia. La unidad de estimación de movimiento 122 puede emitir el índice de imagen de referencia y el vector de movimiento como la información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar el bloque de píxeles predictivo de la PU  
 45 basándose en el bloque de referencia indicado por la información de movimiento de la PU.

**[111]** Si una PU está en un fragmento B, la unidad de estimación de movimiento 122 puede llevar a cabo inter-predicción unidireccional o inter-predicción bidireccional para la PU. Para realizar inter-predicción unidireccional para  
 50 la PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de una primera lista de imágenes de referencia ("lista 0") o una segunda lista de imágenes de referencia ("lista 1") para un bloque de referencia para la PU. La lista 0 y/o la lista 1 pueden incluir una o más imágenes de referencia entre visualizaciones. La unidad de estimación de movimiento 122 puede generar, como la información de movimiento de la PU, un índice de imagen de referencia que indica una posición en la lista 0 o la lista 1 de la imagen de referencia que contiene el  
 55 bloque de referencia, un vector de movimiento que indica un desplazamiento espacial entre el bloque de píxeles de la PU y el bloque de referencia, y un indicador de dirección de predicción que indica si la imagen de referencia está en la lista 0 o en la lista 1.

**[112]** Para realizar inter-predicción bidireccional para una PU, la unidad de estimación de movimiento 122 puede  
 60 buscar las imágenes de referencia en la lista 0 para un bloque de referencia para la PU y también puede buscar las imágenes de referencia de la lista 1 para otro bloque de referencia para la PU. La unidad de estimación de movimiento 122 puede generar índices de imágenes de referencia que indiquen posiciones en la lista 0 y lista 1 de las imágenes de referencia que contienen los bloques de referencia. Además, la unidad de estimación de movimiento 122 puede generar vectores de movimiento que indican desplazamientos espaciales entre los bloques  
 65 de referencia y el bloque de píxeles de la PU. La información de movimiento de la PU puede incluir los índices de imágenes de referencia y los vectores de movimiento de la PU. La unidad de compensación de movimiento 124

puede generar el bloque de píxeles predictivo de la PU basándose en los bloques de referencia indicados por la información de movimiento de la PU

**[113]** La unidad de procesamiento de inter-predicción 121 puede señalar el conjunto de imágenes de referencia (RPS) aplicable a un fragmento. La unidad de procesamiento de inter-predicción 121 puede, de acuerdo con las técnicas de esta divulgación, señalar los componentes de visualización de referencia entre visualizaciones del RPS de varias maneras. Por ejemplo, la unidad de procesamiento de inter-predicción 121 puede señalar los componentes de visualización de referencia entre visualizaciones como se describe en la extensión de MVC. En otro ejemplo, la unidad de procesamiento de inter-predicción 121 puede señalar los componentes de visualización de referencia entre visualizaciones como se describe en la Solicitud de Patente Provisional de los Estados Unidos 61/512 771.

**[114]** En otro ejemplo, la unidad de procesamiento de inter-predicción 121 puede señalar componentes de visualización de referencia entre visualizaciones incluyendo la estructura sintáctica del Tabla 6 siguiente, en un SPS.

**TABLA 6**

|   |       |
|---|-------|
| for(i = 1; i <= num_views_minus1; i++){     |       |
| <b>num_ref_views[i]</b>                     | ue(v) |
| for(j = 0; j < num_ref_views[i]; j++){      |       |
| <b>ref_view_idx[i][j]</b>                   | ue(v) |
| }   |       |
| for(i = 1; i <= num_views_minus1; i++) {    |       |
| <b>num_inter_view_rps[i]</b>                | ue(v) |
| for(k = 0; k < num_inter_view_rps[i]; k++){ |       |
| inter_view_rps(i, k)                        |       |
| }   |       |

**[115]** En la estructura sintáctica de ejemplo de la Tabla 6, los elementos sintácticos num\_ref\_views[i] y los elementos sintácticos ref\_view\_idx[i][j] pueden tener la misma semántica que los de la Solicitud de Patente Provisional de los Estados Unidos 61/512 771. Es decir, el elemento sintáctico "num\_ref\_views[i]" puede especificar el número de componentes de visualización para la predicción entre visualizaciones en las versiones iniciales de la lista 0 y la lista 1 en componentes de visualización de descodificación con índice de orden de visualización igual a i. Los elementos sintácticos "ref\_view\_idx[i][j]" pueden especificar el índice de orden de visualización del componente de visualización j-ésimo en las versiones iniciales de la lista 0 y la lista 1 en la descodificación de un componente de visualización con índice de orden de visualización igual a i. El elemento sintáctico num\_inter\_view\_rps[i] puede especificar el número de las siguientes estructuras sintácticas inter\_view\_rps[i].

**[116]** De forma alternativa, la unidad de procesamiento de inter-predicción 121 puede incluir la sintaxis de la Tabla 7 en un PPS en lugar de un SPS, como en el ejemplo de la Tabla 6.

**TABLA 7**

|   |       |
|---|-------|
| for(i = 1; i <= num_views_minus1; i++) {    |       |
| <b>num_inter_view_rps[i]</b>                | ue(v) |
| for(k = 0; k < num_inter_view_rps[i]; k++){ |       |
| inter_view_rps(i, k)                        |       |
| }   |       |

**[117]** En los ejemplos de las Tablas 6 y 7, la estructura sintáctica "inter\_view\_rps(i, k)" se puede expandir como se muestra en la Tabla 8, a continuación.

**TABLA 8**

|  |      |
|--|------|
| inter_view_rps(i, k)                   |      |
| {                                      |      |
| for(j = 0; j < num_ref_views[i]; j++){ |      |
| <b>ref_view_flag[i][j]</b>             | u(1) |
| }                                      |      |

**[118]** En la estructura sintáctica de ejemplo de la Tabla 8, si el elemento sintáctico "ref\_view\_flag[i][j]" es igual a 1, la unidad de procesamiento de inter-predicción 121 puede incluir un componente de visualización con índice de orden de visualización igual a "ref\_view\_idx[i][ j]" en un conjunto de imágenes de referencia. Si el elemento sintáctico "ref\_view\_flag[i][j]" es igual a 0, el componente de visualización con índice de orden de visualización igual a "ref\_view\_idx[i][j]" no se incluye en el conjunto de imágenes de referencia.

[119] Además, la unidad de procesamiento de inter-predicción 121 puede incluir la estructura sintáctica de la Tabla 9, a continuación, en una cabecera de fragmento.

TABLA 9

5

|                                 |      |
|---------------------------------|------|
| <b>all_inter_view_refs_flag</b> | u(1) |
| if(!all_inter_view_refs_flag) { |      |
| <b>no_inter_view_ref_flag</b>   | u(1) |
| if(!no_inter_view_ref_flag)     |      |
| <b>inter_view_rps_idx</b>       | u(1) |
| }                               |      |

[120] En la sintaxis de ejemplo de la Tabla 9, si el elemento sintáctico "all\_inter\_view\_refs\_flag" es igual a 1, todos los componentes de visualización identificados por los elementos sintácticos "ref\_view\_idx[i][j]" en el SPS están, para cada valor de j, incluidos en el RPS del componente de visualización actual. Si el elemento sintáctico "all\_inter\_view\_refs\_flag" no es igual a 1, el RPS del componente de visualización actual puede incluir un subconjunto de los componentes de visualización identificados por los elementos sintácticos "ref\_view\_idx[i][j]" en el SPS.

[121] Además, en la sintaxis de ejemplo de la Tabla 9, si el elemento sintáctico "all\_inter\_view\_refs\_flag" no es igual a 1, la cabecera de fragmento puede incluir un elemento sintáctico "no\_inter\_view\_ref\_flag". Si el elemento sintáctico "no\_inter\_view\_ref\_flag" es igual a 1, el RPS del componente de visualización actual no incluye ninguna imagen de referencia entre visualizaciones. Si el elemento sintáctico "no\_inter\_view\_ref\_flag" es igual a 0, el RPS del componente de visualización actual puede incluir una o más imágenes de referencia entre visualizaciones. Además, si el elemento sintáctico "no\_inter\_view\_ref\_flag" es igual a 0, la cabecera de fragmento puede incluir un elemento sintáctico "inter\_view\_rps\_idx". El elemento sintáctico "inter\_view\_rps\_idx" puede especificar un índice de una imagen de referencia entre visualizaciones establecida en las estructuras sintácticas "inter\_view\_rps (i, k)" del SPS asociado con el componente de visualización actual. De esta manera, el RPS del componente de visualización actual puede incluir las imágenes de referencia entre visualizaciones en la estructura sintáctica "inter\_view\_rps (i, inter\_view\_rps\_idx)".

[122] La unidad de procesamiento de intra-predicción 126 puede generar datos predictivos para una PU realizando una intra-predicción en la PU. Los datos predictivos para la PU pueden incluir un bloque de píxeles predictivo para la PU y varios elementos sintácticos. La unidad de procesamiento de intra-predicción 126 puede realizar la intra-predicción en PU en fragmentos I, fragmentos P y fragmentos B.

[123] Para realizar la intra-predicción en una PU, la unidad de procesamiento de intra-predicción 126 puede usar múltiples modos de intra-predicción para generar múltiples conjuntos de datos predictivos para la PU. Para usar un modo de intra-predicción para generar un conjunto de datos predictivos para la PU, la unidad de procesamiento de intra-predicción 126 puede extender muestras de bloques de muestras de PU próximas a través del bloques de muestras de la PU en una dirección asociada con el modo de intra-predicción. Las PU próximas pueden estar arriba, arriba y a la derecha, arriba y a la izquierda, o a la izquierda de la PU, asumiendo un orden de codificación de arriba a abajo, de izquierda a derecha para los PU, CU y CTB. La unidad de procesamiento de intra-predicción 126 puede usar varios números de modos de intra-predicción, por ejemplo, 33 modos de intra-predicción direccional. En algunos ejemplos, el número de modos de intra-predicción puede depender del tamaño del bloque de píxeles de la PU.

[124] La unidad de procesamiento de predicción 100 puede seleccionar los datos predictivos para PU de una CU entre los datos predictivos generados por la unidad de procesamiento de inter-predicción 121 para las PU o los datos predictivos generados por la unidad de procesamiento de de intra-predicción 126 para las PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos predictivos para la PU de la CU basándose en las métricas de velocidad / distorsión de los conjuntos de datos predictivos. Los bloques de píxeles predictivos de los datos predictivos seleccionados pueden denominarse en el presente documento bloques de píxeles predictivos seleccionados.

[125] La unidad de generación residual 102 puede generar, basándose en el bloque de píxeles de una CU y los bloques de píxeles predictivos seleccionados de las PU de la CU, un bloque de píxeles residual de una CU. Por ejemplo, la unidad de generación residual 102 puede generar el bloque de píxeles residual de la CU de manera que cada muestra en el bloque de píxeles residual tenga un valor igual a una diferencia entre una muestra en el bloque de píxeles de la CU y una muestra correspondiente en un bloque de píxeles predictivo seleccionado de una PU de la CU.

[126] La unidad de procesamiento de transformación 104 puede realizar divisiones en árbol cuádruple para dividir el bloque de píxeles residual de una CU en sub-bloques. Cada bloque de píxeles residual no dividido puede estar asociado con una TU diferente de la CU. Los tamaños y posiciones de los bloques de píxeles residuales asociados

con las TU de una CU pueden o no basarse en los tamaños y posiciones de los bloques de vídeo asociados con las PU de la CU. Una estructura en árbol cuádruple conocida como "árbol cuádruple residual" (RQT) puede incluir nodos asociados con cada uno de los bloques de píxeles residuales. Las TU de una CU pueden corresponder a nodos hoja del RQT.

5  
**[127]** Dado que los píxeles de los bloques de píxeles residuales de las TU pueden comprender una muestra Y, una muestra U y una muestra V, cada una de las TU puede estar asociada a un bloque de muestras Y, un bloque de muestras U y un bloque de muestras V. Una muestra U puede indicar una diferencia entre un componente azul de un píxel y una muestra Y para el píxel. Por esta razón, una muestra U también se puede denominar muestra Cb.  
 10 Una muestra V puede indicar una diferencia entre una muestra roja de un píxel y una muestra Y para el píxel. Por esta razón, una muestra en V también se puede denominar una muestra Cr.

15  
**[128]** La unidad de procesamiento de transformación 104 puede generar bloques de coeficientes para cada TU de una CU aplicando una o más transformaciones a los bloques de muestras residuales asociados con la TU. La unidad de procesamiento de transformación 104 puede aplicar varias transformaciones a un bloque de muestras residual asociado con una TU. Por ejemplo, la unidad de procesamiento de transformación 104 puede aplicar una transformación de coseno discreta (DCT), una transformación direccional o una transformación conceptualmente similar al bloque de muestras residual.

20  
**[129]** La unidad de procesamiento de transformación 104 puede determinar una transformación para aplicar basándose en un tamaño del bloque de muestras residual al que se aplicará la transformación. Por ejemplo, la unidad de procesamiento de transformación 104 puede aplicar diferentes transformaciones dependiendo de si el bloque de muestras residual es 4x4, 8x8, 16x16, otro tamaño. En algunos ejemplos, la unidad de procesamiento de transformación 104 puede aplicar transformaciones a bloques de muestras residuales de forma rectangular, tales como bloques de muestras residuales que son 16x4, 32x8, etc.  
 25

**[130]** La unidad de cuantificación 106 puede cuantificar los coeficientes en un bloque de coeficientes asociado con una TU. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o a la totalidad de los coeficientes. Por ejemplo, un coeficiente de  $n$  bits puede redondearse a un coeficiente de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ . La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes asociado con una TU de una CU basándose en un valor de parámetro de cuantificación (QP) asociado con la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes asociados con una CU ajustando el valor QP asociado con la CU.  
 30

35  
**[131]** La unidad de cuantificación inversa 108 y la unidad de procesamiento de transformación inversa 110 pueden aplicar cuantificación inversa y transformaciones inversas al bloque de coeficientes, respectivamente, para reconstruir un bloque de muestras residual a partir del bloque de coeficientes. La unidad de reconstrucción 112 puede añadir el bloque de muestras residual reconstruido a las muestras correspondientes de uno o más bloques de muestras predictivos generados por la unidad de procesamiento de predicción 100 para producir un bloque de muestras reconstruido asociado con una TU. Mediante la reconstrucción de bloques de muestras para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir el bloque de píxeles de la CU.  
 40

**[132]** La unidad de filtro 113 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo en el bloque de píxeles asociado con una CU. La memoria intermedia de imágenes descodificadas 114 puede almacenar los bloques de píxeles reconstruidos después de que la unidad de filtro 113 realice la una o más operaciones de desbloqueo en los bloques de píxeles reconstruidos. La unidad de inter-predicción 121 puede utilizar una imagen de referencia que contiene los bloques de píxeles reconstruidos para realizar inter-predicción en las PU de otras imágenes. Además, la unidad de procesamiento de intra-predicción 126 puede usar bloques de píxeles reconstruidos en la memoria intermedia de imágenes descodificadas 114 para realizar intra-predicción en otras PU en la misma imagen que la CU.  
 45  
 50

**[133]** La unidad de codificación por entropía 116 puede recibir datos de otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 116 puede recibir bloques de coeficientes de la unidad de cuantificación 106 y puede recibir elementos sintácticos de la unidad de procesamiento de predicción 100. La unidad de codificación por entropía 116 puede realizar una o más operaciones de codificación por entropía en los datos para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 116 puede realizar una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación de CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de división de intervalo de probabilidad (PIPE), una operación de codificación de Golomb exponencial u otro tipo de operación de codificación por entropía en los datos. El codificador de vídeo 20 puede emitir un flujo de bits que incluye datos codificados por entropía generados por la unidad de codificación por entropía 116.  
 55  
 60

**[134]** La FIG. 5 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo 30 que está configurado para implementar las técnicas de esta divulgación. La FIG. 5 se proporciona con fines de explicación y no se limita a las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta  
 65

divulgación describe el descodificador de vídeo 30 en el contexto de codificación HEVC, y las extensiones de MVC y 3DV de HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

5 **[135]** En el ejemplo de la FIG. 5, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 150, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de procesamiento de transformación inversa 156, una unidad de reconstrucción 158, una unidad de filtro 159 y una memoria intermedia de imágenes descodificadas 160. La unidad de procesamiento de predicción 152 incluye una  
10 unidad de compensación de movimiento 162 y la unidad de procesamiento de intra-predicción 164. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

**[136]** El descodificador de vídeo 30 puede recibir un flujo de bits. La unidad de descodificación por entropía 150 puede analizar el flujo de bits para extraer elementos sintácticos del flujo de bits. Como parte del análisis del flujo de bits, la unidad de descodificación por entropía 150 puede descodificar por entropía de elementos sintácticos  
15 codificados por entropía en el flujo de bits. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de procesamiento de transformación inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 159 pueden generar datos de vídeo descodificados basándose en los elementos sintácticos extraídos del flujo de bits.

20 **[137]** El flujo de bits puede comprender una serie de unidades NAL. Las unidades NAL del flujo de bits pueden incluir unidades NAL de fragmentos codificados. Como parte del análisis del flujo de bits, la unidad de descodificación por entropía 150 puede extraer y descodificar por entropía elementos sintácticos de las unidades NAL de fragmentos codificados. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos sintácticos pertenecientes a un fragmento  
25 de un componente de visualización. Los elementos sintácticos en la cabecera de fragmento pueden incluir un elemento sintáctico que identifica un PPS asociado con una imagen (es decir, un componente de visualización de textura o un componente de visualización de profundidad) que contiene el fragmento.

**[138]** Además, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en una CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Realizando la operación de reconstrucción para cada  
30 TU de la CU, el descodificador de vídeo 30 puede reconstruir un bloque de píxeles residual asociado con la CU.

**[139]** Como parte de la realización de una operación de reconstrucción en una TU de una CU, la unidad de cuantificación inversa 154 puede cuantificar de forma inversa, es decir, descuantificar, los bloques de coeficientes asociados con la TU. La unidad de cuantificación inversa 154 puede utilizar un valor QP asociado con la CU de la TU para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para la unidad de cuantificación inversa 154 a aplicar.  
35

40 **[140]** Después de que la unidad de cuantificación inversa 154 cuantifica inversamente un bloque de coeficientes, la unidad de procesamiento de transformación inversa 156 puede aplicar una o más transformaciones inversas al bloque de coeficientes para generar un bloque de muestras residual asociado con la TU. Por ejemplo, la unidad de procesamiento de transformación inversa 156 puede aplicar una DCT inversa, una transformación de número entero inversa, una transformación de Karhunen-Loeve (KLT) inversa, una transformación de rotación inversa, una  
45 transformación direccional inversa u otra transformación inversa al bloque de coeficientes.

**[141]** Si se codifica una PU usando intra-predicción, la unidad de procesamiento de intra-predicción 164 puede realizar intra-predicción para generar un bloque de muestras predictivo para la PU. La unidad de procesamiento de intra-predicción 164 puede usar un modo de intra-predicción para generar el bloque de píxeles predictivo para la PU basándose en los bloques de píxeles de PU espacialmente contiguas. La unidad de procesamiento de intra-predicción 164 puede determinar el modo de intra-predicción para la PU basándose en uno o más elementos sintácticos analizados desde el flujo de bits.  
50

**[142]** Cuando el descodificador de vídeo 30 descodifica un fragmento P o B, la unidad de procesamiento de predicción 152 puede generar, basándose en elementos sintácticos extraídos del flujo de bits, una primera lista de imágenes de referencia (lista 0) y una segunda lista de imágenes de referencia (lista 1) para el fragmento P o B. Si se codifica una PU usando la inter-predicción, la unidad de compensación de movimiento 162 puede generar, basándose en uno o más bloques de referencia para la PU, un bloque de píxeles predictivo para la PU. Para determinar un bloque de referencia para la PU, la unidad de compensación de movimiento 162 puede determinar, basándose en un índice de imagen de referencia de la PU, una imagen de referencia en una lista de imágenes de referencia para la trama asociada con la PU. Además, la unidad de compensación de movimiento 162 puede identificar, basándose en un vector de movimiento de la PU, un bloque de referencia en la imagen de referencia identificada.  
55 60

65 **[143]** Para generar la lista 0 y la lista 1, la unidad de procesamiento de predicción 152 puede realizar un proceso de inicialización para la lista 0 y un proceso de inicialización para la lista 1. En diversos ejemplos, la unidad de

procesamiento de predicción 152 puede realizar diversos procesos de inicialización.

**[144]** En un ejemplo, la unidad de procesamiento de predicción 152 puede insertar imágenes de referencia entre visualizaciones en la lista 0 inmediatamente después de las imágenes de referencia de RefPicSetStCurr0. En este ejemplo, la unidad de procesamiento de predicción 152 puede insertar imágenes de referencia entre visualizaciones en la lista 1 inmediatamente después de las imágenes de referencia de RefPicSetCurr1. En este proceso de inicialización de ejemplo, tanto la lista 0 como la lista 1 incluyen al menos una imagen de referencia de RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetLtCurr o RefPicSetInterView. El siguiente pseudocódigo describe un proceso de inicialización de ejemplo para la lista 0.

```

10  cldx = 0

    while(cldx <= num_ref_idx_10_active_minus1)
15  {
        for(i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
            RefPicList0[cldx] = pic(RefPicSetStCurr0[i], VOIdx)
20  for(i=0; i < NumRpsInterView && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
            RefPicList0[cldx] = pic(PicOrderCnt, RefPicSetInterView[i])
25  for(i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
            RefPicList0[cldx] = pic(RefPicSetStCurr1[i], VOIdx)
30  for(i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
            RefPicList0[cldx] = pic(RefPicSetLtCurr[i], VOIdx)
    }

```

**[145]** En el pseudocódigo anterior, cldx es una variable de contador, NumRpsStCurr0 es el número de imágenes de referencia en RefPicSetStCurr0, NumRpsStCurr1 es el número de imágenes de referencia en RefPicSetStCurr1, NumRpsLtCurr es el número de imágenes de referencia en RefPicSetLtCurr, RefPicList0 es la lista 0 y RefPicList1 es la lista 1. Además, RefPicSet-InterView es el conjunto de imágenes de referencia entre visualizaciones. La función pic() devuelve una imagen de referencia, dado un valor POC y un índice de orden de visualización. VOIdx es un índice de orden de visualización del componente de visualización actual. PicOrderCnt es un valor POC del componente de visualización actual. De forma alternativa, la función pic () devuelve una imagen de referencia, dado un valor POC de la imagen de referencia y un identificador de visualización (viewId) de la imagen de referencia. Además, num\_ref\_idx\_l0\_active\_minus1 indica el número de imágenes de referencia activas en la lista 0, menos 1. num\_ref\_idx\_l1\_active\_minus1 indica el número de imágenes de referencia activas en la lista 1, menos 1.

**[146]** Cuando la unidad de procesamiento de predicción 152 ejecuta el pseudocódigo anterior, la unidad de procesamiento de predicción 152 inserta, en la lista de imágenes de referencia 0, cada imagen de referencia de RefPicSetStCurr0 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetInterView con un valor POC igual a un valor POC del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetStCurr1 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetLtCurr con el índice de orden de visualización del componente de visualización actual.

**[147]** El siguiente pseudocódigo describe un proceso de inicialización de ejemplo para la lista 1.

```

55  cldx = 0

    while(cldx <= num_ref_idx_11_active_minus1)
60  {
        for(i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
            RefPicList1[cldx] = pic(RefPicSetStCurr1[i], VOIdx)
65  for(i=0; i < NumRpsInterView && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)

```

```

RefPicList1[cldx] = pic(PicOrderCnt, RefPicSetInterView[i])
5   for(i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
      RefPicList1[cldx] = pic(RefPicSetStCurr0[i], VOIdx)
      for(i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
10     RefPicList1 [cldx] = pic (RefPicSetLtCurr [i], VOIdx)
}

```

15 Cuando la unidad de procesamiento de predicción 152 ejecuta el pseudocódigo anterior, la unidad de procesamiento de predicción 152 inserta, en la lista de imágenes de referencia 1, cada imagen de referencia de RefPicSetStCurr1 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetInterView con un valor POC igual a un valor POC del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetStCurr0 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetLtCurr con el índice de orden de visualización del componente de visualización actual.

20 **[148]** En otro ejemplo de proceso de inicialización de la lista de imágenes de referencia, la unidad de procesamiento de predicción 152 puede insertar imágenes de referencia entre visualizaciones en la lista 0 inmediatamente después de las imágenes de referencia de RefPicSetStCurr1. En este ejemplo, la unidad de procesamiento de predicción 152 puede insertar imágenes de referencia entre visualizaciones en la lista 1 inmediatamente después de las imágenes de referencia de RefPicSetStCurr0. En este proceso de inicialización de ejemplo, tanto la lista 0 como la lista 1 incluyen al menos una imagen de referencia de RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetLtCurr o RefPicSetInterView. El siguiente pseudocódigo describe este proceso de inicialización de ejemplo para la lista 0.

```

30   cldx = 0
      while(cldx <= num_ref_idx_10_active_minus1)
35   {
      for(i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
          RefPicList0[cldx] = pic(RefPicSetStCurr0[i], VOIdx)
40   for(i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
          RefPicList0[cldx] = pic(RefPicSetStCurr1[i], VOIdx)
45   for(i=0; i < NumRpsInterView && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
          RefPicList0[cldx] = pic(PicOrderCnt, RefPicSetInterView[i])
      for(i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
50     RefPicList0[cldx] = pic(RefPicSetLtCurr[i], VOIdx)
}

```

55 **[149]** Cuando la unidad de procesamiento de predicción 152 ejecuta el pseudocódigo anterior, la unidad de procesamiento de predicción 152 inserta, en la lista de imágenes de referencia 0, cada imagen de referencia de RefPicSetStCurr0 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetStCurr1 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetInterView con un valor POC igual a un valor POC del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetLtCurr con el índice de orden de visualización del componente de visualización actual.

60 **[150]** El siguiente pseudocódigo describe un proceso de inicialización de ejemplo para la lista 1.

```

65   cldx = 0

```

```

while(cldx <= num_ref_idx_11_active_minus1)
{
5       for(i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
           RefPicList1[cldx] = pic(RefPicSetStCurr1[i], VOIdx)
           for(i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
10          RefPicList1[cldx] = pic(RefPicSetStCurr0[i], VOIdx)
           for(i=0; i < NumRpsInterView && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
15          RefPicList1[cldx] = pic(PicOrderCnt, RefPicSetInterView[i])
           for(i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
20          RefPicList1 [cldx] = pic (RefPicSetLtCurr [i], VOIdx)
}

```

Los términos en el pseudocódigo anterior pueden tener los mismos significados que los términos en el pseudocódigo del ejemplo anterior. Cuando la unidad de procesamiento de predicción 152 ejecuta el pseudocódigo anterior, la unidad de procesamiento de predicción 152 inserta, en la lista de imágenes de referencia 1, cada imagen de referencia de RefPicSetStCurr0 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetStCurr1 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetInterView con un valor POC igual a un valor POC del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetLtCurr con el índice de orden de visualización del componente de visualización actual.

**[151]** En otro ejemplo de proceso de inicialización de la lista de imágenes de referencia, la unidad de procesamiento de predicción 152 puede insertar imágenes de referencia entre visualizaciones en la lista 0 y en la lista 1 inmediatamente después de las imágenes de referencia de RefPicSetLtCurr. En este proceso de inicialización de ejemplo, tanto la lista 0 como la lista 1 incluyen al menos una imagen de referencia de RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetLtCurr o RefPicSetInterView. El siguiente pseudocódigo describe este proceso de inicialización de ejemplo para la lista 0.

```

cldx = 0
40 while(cldx <= num_ref_idx_10_active_minus1)
{
45       for(i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
           RefPicList0[cldx] = pic(RefPicSetStCurr0[i], VOIdx)
           for(i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
50          RefPicList0[cldx] = pic(RefPicSetStCurr1[i], VOIdx)
           for(i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
55          RefPicList0[cldx] = pic(RefPicSetLtCurr[i], VOIdx)
           for(i=0; i < NumRpsInterView && cldx <= num_ref_idx_10_active_minus1; cldx++, i++)
60          RefPicList0[cldx] = pic(PicOrderCnt, RefPicSetInterView[i])
}

```

**[152]** Cuando la unidad de procesamiento de predicción 152 ejecuta el pseudocódigo anterior, la unidad de procesamiento de predicción 152 inserta, en la lista de imágenes de referencia 0, cada imagen de referencia de RefPicSetStCurr0 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetStCurr1 con el índice de orden de visualización del componente de visualización

actual, seguido de cada imagen de referencia de RefPicSetLtCurr con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetInterView con un valor POC igual a un valor POC del componente de visualización actual.

5 **[153]** El siguiente pseudocódigo describe este proceso de inicialización de ejemplo para la lista 1.

```

cldx = 0
while(cldx <= num_ref_idx_11_active_minus1)
10 {
    for(i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
15     RefPicList1[cldx] = pic(RefPicSetStCurr1[i], VOIdx)
    for(i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
        RefPicList1[cldx] = pic(RefPicSetStCurr0[i], VOIdx)
20    for(i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
        RefPicList1[cldx] = pic(RefPicSetLtCurr[i], VOIdx)
25    for(i=0; i < NumRpsInterView && cldx <= num_ref_idx_11_active_minus1; cldx++, i++)
        RefPicList1[cldx] = pic(PicOrderCnt, RefPicSetInterView[i])
30 }

```

Los términos en el pseudocódigo anterior pueden tener los mismos significados que los términos en el pseudocódigo del ejemplo anterior. Cuando la unidad de procesamiento de predicción 152 ejecuta el pseudocódigo anterior, la unidad de procesamiento de predicción 152 inserta, en la lista de imágenes de referencia 1, cada imagen de referencia de RefPicSetStCurr1 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetStCurr0 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetLtCurr con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetInterView con un valor POC igual a un valor POC del componente de visualización actual.

40 **[154]** En otro ejemplo de proceso de inicialización de lista de imágenes de referencia, la unidad de procesamiento de predicción 152 puede insertar imágenes de referencia entre visualizaciones en la lista 0 inmediatamente después de las imágenes de referencia de RefPicSetStCurr0, RefPicSetStCurr1, o RefPicSetLtCurr como se describe en los ejemplos anteriores. Asimismo, en este ejemplo, la unidad de procesamiento de predicción 152 puede insertar imágenes de referencia entre visualizaciones en la lista 1 inmediatamente después de las imágenes de referencia de RefPicSetStCurr1, RefPicSetStCurr0, o RefPicSetLtCurr como se describe en los ejemplos previos. Sin embargo, en este ejemplo, la unidad de procesamiento de predicción 152 puede insertar en la lista 0 imágenes de referencia entre visualizaciones con identificadores de visualización más pequeños que el identificador de visualización del componente de visualización actual. Además, en este ejemplo, la unidad de procesamiento de predicción 152 puede insertar en la lista 1 imágenes de referencia entre visualizaciones con identificadores de visualización mayores que el identificador de visualización del componente de visualización actual. En este proceso de inicialización de ejemplo, tanto la lista 0 como la lista 1 incluyen al menos una imagen de referencia de RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetLtCurr o RefPicSetInterView. El siguiente pseudocódigo describe este proceso de inicialización de ejemplo para la lista 0.

```

55 cldx = 0
while( cldx <= num_ref_idx_10_active_minus1 )
{
60     for( i=0; i < NumRpsStCurrO && cldx <= num_ref_idx_10_active_minus1; cldx++, i++ )
        RefPicList0[ cldx ] = pic( RefPicSetStCurr0[ i ], VOIdx )
65     for( i=0; i < NumRpsInterViewFwd && cldx <= num_ref_idx_10_active_minus1; cldx++, i++ )

```

```

RefPicList0[ cldx ] = pic( PicOrderCnt, RefPicSetInterViewFwd[ i ] )
for( i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++ )
5   RefPicList0[ cldx ] = pic( RefPicSetStCurr1[ i ], VOIdx )
for( i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_10_active_minus_1; cldx++, i++ )
10  RefPicList0[ cldx ] = pic( RefPicSetLtCurr[ i ], VOIdx )
}

```

**[155]** Cuando la unidad de procesamiento de predicción 152 ejecuta el pseudocódigo anterior, la unidad de procesamiento de predicción 152 inserta, en la lista de imágenes de referencia 0, cada imagen de referencia de RefPicSetStCurr0 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetInterViewFwd con un valor POC igual a un valor POC del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetStCurr1 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetLtCurr con el índice de orden de visualización del componente de visualización actual.

**[156]** El siguiente pseudocódigo describe este proceso de inicialización de ejemplo para la lista 1.

```

cldx = 0
25 while( cldx <= num_ref_idx_11_active_minus1 )
{
30   for( i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++ )
       RefPicList1[ cldx ] = pic( RefPicSetStCurr1[ i ], VOIdx )
   for( i=0; i < NumRpsInterViewBwd && cldx <= num_ref_idx_11_active_minus1; cldx++, i++ )
35     RefPicList1[ cldx ] = pic( PicOrderCnt, RefPicSetInterViewBwd[ i ] )
   for( i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++ )
       RefPicList1[ cldx ] = pic( RefPicSetStCurr0[ i ], VOIdx )
40   for( i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_11_active_minus1; cldx++, i++ )
       RefPicList1[ cldx ] = pic( RefPicSetLtCurr[ i ], VOIdx )
45 }

```

En el pseudocódigo anterior, RefPicSetInterViewFwd es el subconjunto de imágenes de referencia entre visualizaciones con identificadores de visualización más pequeños que el identificador de visualización del componente de visualización actual. RefPicSetInterViewBwd es el subconjunto de imágenes de referencia entre visualizaciones con identificadores de visualización mayores que el identificador de visualización del componente de visualización actual. Los otros términos en el pseudocódigo anterior pueden tener los mismos significados que los términos en el pseudocódigo de los ejemplos anteriores.

**[157]** Cuando la unidad de procesamiento de predicción 152 ejecuta el pseudocódigo anterior, la unidad de procesamiento de predicción 152 inserta, en la lista de imágenes de referencia 1, cada imagen de referencia de RefPicSetStCurr1 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetInterViewBwd con un valor POC igual a un valor POC del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetStCurr0 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetLtCurr con el índice de orden de visualización del componente de visualización actual.

**[158]** En otro ejemplo de proceso de inicialización de lista de imágenes de referencia, la unidad de procesamiento de predicción 152 puede insertar imágenes de referencia entre visualizaciones en la lista 0 inmediatamente después de las imágenes de referencia de RefPicSetStCurr0, RefPicSetStCurr1, o RefPicSetLtCurr como se describe en los ejemplos anteriores. Asimismo, en este ejemplo, la unidad de procesamiento de predicción 152 puede insertar imágenes de referencia entre visualizaciones en la lista 1 inmediatamente después de las imágenes de referencia de

RefPicSetStCurr1, RefPicSetStCurr0, o RefPicSetLtCurr como se describe en los ejemplos previos. Sin embargo, cuando la unidad de procesamiento de predicción 152 incluye componentes de visualización de referencia entre visualizaciones en la lista de imágenes de referencia 0, la unidad de procesamiento de predicción 152 puede incluir cada componente de visualización de referencia entre visualizaciones de RefPicSetInterViewFwd antes de cualquier componente de visualización de referencia entre visualizaciones de RefPicSetInterViewBwd. Cuando la unidad de procesamiento de predicción 152 incluye componentes de visualización de referencia entre visualizaciones en la lista de imágenes de referencia 1, la unidad de procesamiento de predicción 152 puede incluir cada componente de visualización de referencia entre visualizaciones de RefPicSetInterViewBwd antes de los componentes de visualización de referencia entre visualizaciones de RefPicSetInterViewFwd. Como en el ejemplo anterior, RefPicSetInterViewFwd es el subconjunto de imágenes de referencia entre visualizaciones con identificadores de visualización más pequeños que el identificador de visualización del componente de visualización actual y RefPicSetInterViewBwd es el subconjunto de imágenes de referencia entre visualizaciones con identificadores de visualización más grandes que el identificador de visualización del componente de visualización actual. En este proceso de inicialización de ejemplo, tanto la lista 0 como la lista 1 incluyen al menos una imagen de referencia de RefPicSetStCurr0, RefPicSetStCurr1, RefPic-SetLtCurr o RefPicSetInterView. El siguiente pseudocódigo describe este proceso de inicialización de ejemplo para la lista 0.

```
cldx = 0
```

```
while( cldx <= num_ref_idx_10_active_minus1 )
```

```
{
```

```
    for( i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++ )
```

```
        RefPicList0[ cldx ] = pic( RefPicSetStCurr0[ i ], VOIdx )
```

```
    for( i=0; i < NumRpsInterViewFwd && cldx <= num_ref_idx_10_active_minus1; cldx++, i++ )
```

```
        RefPicList0[ cldx ] = pic( PicOrderCnt, RefPicSetInterViewFwd[ i ] )
```

```
    for( i=0; i < NumRpsInterViewBwd && cldx <= num_ref_idx_10_active_minus1; cldx++, i++ )
```

```
        RefPicList0[ cldx ] = pic( PicOrderCnt, RefPicSetInterViewBwd[ i ] )
```

```
    for( i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_10_active_minus1; cldx++, i++ )
```

```
        RefPicList0[ cldx ] = pic( RefPicSetStCurr1[ i ], VOIdx )
```

```
    for( i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_10_active_minus1; cldx++, i++ )
```

```
        RefPicList0[ cldx ] = pic( RefPicSetLtCurr[ i ], VOIdx )
```

```
}
```

**[159]** Cuando la unidad de procesamiento de predicción 152 ejecuta el pseudocódigo anterior, la unidad de procesamiento de predicción 152 inserta, en la lista de imágenes de referencia 0, cada imagen de referencia de RefPicSetStCurr0 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetInterViewFwd con un valor POC igual al valor POC del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetStCurr1 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetLtCurr con el índice de orden de visualización del componente de visualización actual.

**[160]** El siguiente pseudocódigo describe este proceso de inicialización de ejemplo para la lista 1.

```
cldx = 0
```

```
while( cldx <= num_ref_idx_11_active_minus1 )
```

```
{
```

```
    for( i=0; i < NumRpsStCurr1 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++ )
```

```
        RefPicList1[ cldx ] = pic( RefPicSetStCurr1[ i ], VOIdx )
```

```

for( i=0; i < NumRpsInterViewBwd && cldx <= num_ref_idx_11_active_minus1; cldx++, i++ )
    RefPicList1[ cldx ] = pic( PicOrderCnt, RefPicSetInterViewBwd[ i ] )
5   for( i=0; i < NumRpsInterViewFwd && cldx <= num_ref_idx_11_active_minus1; cldx++, i++ )
    RefPicList1[ cldx ] = pic( PicOrderCnt, RefPicSetInterViewFwd[ i ] )

for( i=0; i < NumRpsStCurr0 && cldx <= num_ref_idx_11_active_minus1; cldx++, i++ )
10  RefPicList1[ cldx ] = pic( RefPicSetStCurr0[ i ], VOIdx )

for( i=0; i < NumRpsLtCurr && cldx <= num_ref_idx_11_active_minus1; cldx++, i++ )
15  RefPicList1[ cldx ] = pic( RefPicSetLtCurr[ i ], VOIdx )
}

```

20 Los términos en el pseudocódigo anterior pueden tener los mismos significados que los términos en el pseudocódigo del ejemplo anterior. Cuando la unidad de procesamiento de predicción 152 ejecuta el pseudocódigo anterior, la unidad de procesamiento de predicción 152 inserta, en la lista de imágenes de referencia 1, cada imagen de referencia de RefPicSetStCurr1 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetInterViewBwd con un valor POC igual al valor POC del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetStCurr0 con el índice de orden de visualización del componente de visualización actual, seguido de cada imagen de referencia de RefPicSetLtCurr con el índice de orden de visualización del componente de visualización actual.

30 **[161]** En algunos procesos de inicialización de la lista de imágenes de referencia de ejemplo, la unidad de procesamiento de predicción 152 puede inicializar las listas 0 y 1 para incluir VSRP. La unidad de procesamiento de predicción 152 puede incluir los VSRP en diversas posiciones dentro de las listas de imágenes de referencia 0 y 1. Por ejemplo, la unidad de procesamiento de predicción 152 puede inicializar las listas de imágenes de referencia 0 y 1 de manera que los VSRP sean inmediatamente posteriores a los componentes de visualización de referencia entre visualizaciones. En otro ejemplo, la unidad de procesamiento de predicción 152 puede inicializar las listas de imágenes de referencia 0 y 1 de manera que las VSPR estén inmediatamente antes de los componentes de visualización de referencia entre visualizaciones.

40 **[162]** Después de que la unidad de procesamiento de predicción 152 realice el proceso de inicialización en la lista 0 y lista 1, la unidad de procesamiento de predicción 152 puede realizar un proceso de modificación de la lista de imágenes de referencia para modificar el orden de las imágenes de referencia en la lista 0 y/o la lista 1. La unidad de procesamiento de predicción 152 puede modificar el orden de las imágenes de referencia en la lista 0 y la lista 1 en respuesta a los comandos RPLM señalados en el flujo de bits. Como se analizó anteriormente, los comandos RPLM pueden incluir elementos sintácticos "list\_modification\_idx". Los comandos RPLM pueden tener varias sintaxis.

45 **[163]** En una primera sintaxis de ejemplo para comandos RPLM, si la unidad de procesamiento de predicción 152 modifica la lista 0 y el elemento sintáctico "list\_modification\_idx" es igual a 0, el elemento sintáctico "ref\_pic\_set\_idx" está presente y el elemento sintáctico "ref\_pic\_set\_idx" corresponde a un índice para una posición en RefPicSetStCurr0. Si la unidad de procesamiento de predicción 152 modifica la lista 1 y el elemento sintáctico "list\_modification\_idx" es igual a 0, el elemento sintáctico "ref\_pic\_set\_idx" está presente y el elemento sintáctico "ref\_pic\_set\_idx" corresponde a un índice a una posición en RefPicSetStCurr1. Además, si la unidad de procesamiento de predicción 152 modifica la lista 0 y el elemento sintáctico "list\_modification\_idx" es igual a 1, el elemento sintáctico "ref\_pic\_set\_idx" está presente y el elemento sintáctico "ref\_pic\_set\_idx" corresponde a un índice a una posición en RefPicSetStCurr1. Si la unidad de procesamiento de predicción 152 modifica la lista 1 y el elemento sintáctico "list\_modification\_idx" es igual a 1, el elemento sintáctico "ref\_pic\_set\_idx" está presente y el elemento sintáctico "ref\_pic\_set\_idx" corresponde a un índice a una posición en RefPicSetStCurr0. Si el elemento sintáctico "list\_modification\_idx" es igual a 2, el elemento sintáctico "ref\_pic\_set\_idx" está presente y el elemento sintáctico "ref\_pic\_set\_idx" indica un índice a una posición en RefPicSetLtCurr. Si el elemento sintáctico "list\_modification\_idx" es igual a 3, la unidad de procesamiento de predicción 152 detiene el ciclo de modificación de la lista de imágenes de referencia.

60 **[164]** En esta primera sintaxis de ejemplo, el elemento sintáctico "list\_modification\_idx" también puede ser igual a 4 o 5. Si el elemento sintáctico "list\_modification\_idx" es igual a 4, el elemento sintáctico "ref\_pic\_set\_idx" está presente y corresponde a un índice de RefPicSetInterView. Si el elemento sintáctico "list\_modification\_idx" es igual a 5, el elemento sintáctico "ref\_pic\_set\_idx" está presente y corresponde a un índice de RefPicSetVsrp. RefPicSetVsrp puede ser un conjunto de VSRP que el componente de visualización actual puede utilizar para la referencia de predicción. En este ejemplo, el elemento sintáctico "ref\_pic\_set\_idx" puede especificar un índice para RefPicSetStCurr0, RefPicSetCurr1, RefPicSetLtCurr, RefPicSetInterView o RefPicSetVsrp de la imagen de referencia

que se mueve al índice actual (por ejemplo, refIdxLX) en la lista de imágenes de referencia.

[165] En un segundo ejemplo de sintaxis para comandos RPLM similar al primer ejemplo de sintaxis descrito en el párrafo anterior, nunca puede haber más de un VSRP presente. Dado que solo hay un VSPR presente, no es necesario indicar qué VSRP insertar en una lista de imágenes de referencia. Por lo tanto, en este ejemplo, el elemento sintáctico "ref\_pic\_set\_idx" no se señala si el elemento sintáctico "list\_modification\_idc" es igual a 5. Además, puede no ser necesario señalar un elemento sintáctico "list\_modification\_idc" igual a 5 para la extensión de múltiples visualizaciones de HEVC.

[166] En una tercera sintaxis de ejemplo para comandos RPLM, el conjunto de imágenes de referencia entre visualizaciones se divide en RefPic-SetInterViewFwd y RefPicSetInterViewBwd. RefPicSetInterViewFwd es un conjunto de imágenes de referencia entre visualizaciones con identificadores de visualización más pequeños que un identificador de visualización del componente de visualización actual. RefPicSetInterViewBwd es un conjunto de imágenes de referencia entre visualizaciones con identificadores de visualización mayores que el identificador de visualización del componente de visualización actual. En este ejemplo, el elemento sintáctico "list\_modification\_idc" también puede ser igual a 4, 5 o 6. Si el elemento sintáctico "list\_modification\_idc" es igual a 4, el elemento sintáctico "ref\_pic\_set\_idx" está presente y corresponde a un índice de una posición en RefPicSetInter-ViewFwd. Si el elemento sintáctico "list\_modification\_idc" es igual a 5, el elemento sintáctico "ref\_pic\_set\_idx" está presente y corresponde a un índice de una posición en RefPicSetInterViewBwd. Si el elemento sintáctico "list\_modification\_idc" es igual a 6, el elemento sintáctico "ref\_pic\_set\_idx" está presente y corresponde a un índice de una posición en RefPicSetVsrp.

[167] En una cuarta sintaxis de ejemplo para los comandos de RPLM, el elemento sintáctico "ref\_pic\_set\_idx" especifica un valor relativo de un índice. Es decir, el elemento sintáctico "ref\_pic\_set\_idx" indica una diferencia entre un índice y un valor previamente señalado del elemento sintáctico "ref\_pic\_set\_idx". En este ejemplo, los comandos RPLM pueden ajustarse a la sintaxis de la Tabla 10 siguiente.

**TABLA 10**

| ref_pic_list_modification() {            | Descriptor |
|--|------------|
| if( slice_type!= 2 ) {                   |            |
| <b>ref_pic_list_modification_flag_10</b> | u(1)       |
| if( ref_pic_list_modification_flag_10 )  |            |
| do {                                     |            |
| <b>list_modification_idc</b>             | ue(v)      |
| if( list_modification_idc != 3 )         |            |
| <b>ref_pic_set_idx_diff_minus1</b>       | ue(v)      |
| } while( list_modification_idc != 3 )    |            |
| }  |            |
| if( slice_type % 5 == 1 ) {              |            |
| <b>ref_pic_list_modification_flag_11</b> | u(1)       |
| if( ref_pic_list_modification_flag_11 )  |            |
| do {                                     |            |
| <b>list_modification_idc</b>             | ue(v)      |
| if( list_modification_idc != 3 )         |            |
| <b>ref_pic_set_idx_diff_minus1</b>       | ue(v)      |
| } while( list_modification_idc != 3 )    |            |
| }  |            |
| }  |            |

[168] En la sintaxis de ejemplo de la Tabla 10, el elemento sintáctico "list\_modification\_idc" puede tener un valor entre 0 y 10. Si el elemento sintáctico "list\_modification\_idc" es igual a 3, la unidad de procesamiento de predicción 152 finaliza el ciclo de modificación de una lista de imágenes de referencia. En los ejemplos siguientes, preRefPicSetIdx es un valor de un elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" anterior. preRefPicSetIdx puede ser inicialmente igual a -1. MaxNum es el número máximo de imágenes de referencia en el subconjunto de imágenes de referencia relevante (por ejemplo, RefPicSetStCurr0, RefPicSetStCurr1, etc.).

[169] Si la unidad de procesamiento de predicción 152 modifica la lista 0 y el elemento sintáctico "list\_modification\_idc" es igual a 0, el elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" está presente y la unidad de procesamiento de predicción 152 puede determinar un índice de una posición en RefPicSetStCurr0 calculando preRefPicSetIdx - (ref\_pic\_set\_idx\_diff\_minus1 + 1) % MaxNum. Si la unidad de procesamiento de predicción 152 modifica la lista 0 y el elemento sintáctico "list\_modification\_idc" es igual a 1, el elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" está presente y la unidad de procesamiento de predicción 152 puede determinar un índice de una posición en RefPicSetStCurr1 calculando preRefPicSetIdx + (ref\_pic\_set\_idx\_diff\_minus1 + 1) % MaxNum.

- 5 **[170]** Si la unidad de procesamiento de predicción 152 modifica la lista 0 y el elemento sintáctico "list\_modification\_idc" es igual a 2, el elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" está presente y la unidad de procesamiento de predicción 152 puede determinar un índice de una posición en RefPicSetStCurr1 calculando  $\text{preRefPicSetIdx} - (\text{ref\_pic\_set\_idx\_diff\_minus1} + 1) \% \text{MaxNum}$ . Si la unidad de procesamiento de predicción 152 modifica la lista 1 y el elemento sintáctico "list\_modification\_idc" es igual a 4, el elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" está presente y la unidad de procesamiento de predicción 152 puede determinar un índice de una posición en RefPicSetStCurr0 calculando  $\text{preRefPicSetIdx} + (\text{ref\_pic\_set\_idx\_diff\_minus1} + 1) \% \text{MaxNum}$ .
- 10 **[171]** Si la unidad de procesamiento de predicción 152 modifica la lista 0 o lista 1 y el elemento sintáctico "list\_modification\_idc" es igual a 5, el elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" está presente y la unidad de procesamiento de predicción 152 puede determinar un índice de una posición en RefPicSetLtCurr calculando  $\text{preRefPicSetIdx} - (\text{ref\_pic\_set\_idx\_diff\_minus1} + 1) \% \text{MaxNum}$ . Si la unidad de procesamiento de predicción 152 modifica la lista 0 o lista 1 y el elemento sintáctico "list\_modification\_idc" es igual a 6, el elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" está presente y la unidad de procesamiento de predicción 152 puede determinar un índice de una posición en RefPicSetLtCurr calculando  $\text{preRefPicSetIdx} + (\text{ref\_pic\_set\_idx\_diff\_minus1} + 1) \% \text{MaxNum}$ .
- 15 **[172]** Si la unidad de procesamiento de predicción 152 modifica la lista 0 o lista 1 y el elemento sintáctico "list\_modification\_idc" es igual a 7, el elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" está presente y la unidad de procesamiento de predicción 152 puede determinar un índice de una posición en RefPicSetInterView calculando  $\text{preRefPicSetIdx} - (\text{ref\_pic\_set\_idx\_diff\_minus1} + 1) \% \text{MaxNum}$ . Si la unidad de procesamiento de predicción 152 modifica la lista 0 o lista 1 y el elemento sintáctico "list\_modification\_idc" es igual a 8, el elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" está presente y la unidad de procesamiento de predicción 152 puede determinar un índice de una posición en RefPicSetInterView calculando  $\text{preRefPicSetIdx} + (\text{ref\_pic\_set\_idx\_diff\_minus1} + 1) \% \text{MaxNum}$ .
- 20 **[173]** Si la unidad de procesamiento de predicción 152 modifica la lista 0 o lista 1 y el elemento sintáctico "list\_modification\_idc" es igual a 9, el elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" está presente y la unidad de procesamiento de predicción 152 puede determinar un índice de una posición en RefPicSetInterVsrp calculando  $\text{preRefPicSetIdx} - (\text{ref\_pictset\_idx\_diff\_minus1} + 1) \% \text{MaxNum}$ . Si la unidad de procesamiento de predicción 152 modifica la lista 0 o lista 1 y el elemento sintáctico "list\_modification\_idc" es igual a 10, el elemento sintáctico "ref\_pic\_set\_idx\_diff\_minus1" está presente y la unidad de procesamiento de predicción 152 puede determinar un índice de una posición en RefPicSetInterVsrp calculando  $\text{preRefPicSetIdx} + (\text{ref\_pic\_set\_idx\_diff\_minus1} + 1) \% \text{MaxNum}$ .
- 25 **[174]** La unidad de procesamiento de predicción 152 puede realizar diversos procesos de modificación de la lista de imágenes de referencia en función de los comandos RPLM. En un ejemplo, la unidad de procesamiento de predicción 152 puede recibir elementos sintácticos que se ajusten a las sintaxis RPLM del primer o segundo ejemplo anteriores. En este ejemplo, cuando el elemento sintáctico "ref\_pic\_list\_modification\_flag\_10" es igual a 1, la unidad de procesamiento de predicción 152 puede procesar los elementos sintácticos "list\_modification\_idc" en un orden en que los elementos sintácticos "list\_modification\_idc" ocurren en el flujo de bits. Para cada elemento sintáctico "list\_modification\_idc", la unidad de procesamiento de predicción 152 puede determinar si el elemento sintáctico "list\_modification\_idc" es igual a 0, 1, 2, 4 o 5. Si el elemento sintáctico "list\_modification\_idc" es igual a 0, 1, 2, 4 o 5, la unidad de procesamiento de predicción 152 puede realizar un proceso para mover una imagen de referencia a una posición en la lista 0 indicada por el índice de imagen de referencia. La unidad de procesamiento de predicción 152 puede proporcionar un índice reldxL0 como entrada al proceso y puede establecer reldxL0 a la salida del proceso. RefldxL0 es un índice en la lista 0. La unidad de procesamiento de predicción 152 puede establecer inicialmente que reldxL0 sea igual a 0. El proceso para mover una imagen de referencia a una posición en la lista 0 indicada por el índice de imagen de referencia se describe a continuación. De lo contrario, si el elemento sintáctico "list\_modification\_idc" es igual a 3, la unidad de procesamiento de predicción 152 detiene el proceso de modificación de la lista de imágenes de referencia para la lista 0.
- 30 **[175]** En este ejemplo, cuando un fragmento actual es un fragmento B y el elemento sintáctico "ref\_pic\_list\_modification\_flag\_11" es igual a 1, la unidad de procesamiento de predicción 152 puede procesar los elementos sintácticos "list\_modification\_idc" en un orden en que los elementos sintácticos "list\_modification\_idc" ocurren en el flujo de bits. Para cada elemento sintáctico "list\_modification\_idc", la unidad de procesamiento de predicción 152 puede determinar si el elemento sintáctico "list\_modification\_idc" es igual a 0, 1, 2, 4 o 5. Si el elemento sintáctico "list\_modification\_idc" es igual a 0, 1, 2, 4 o 5, la unidad de procesamiento de predicción 152 puede realizar un proceso para mover una imagen de referencia a una posición en la lista 1 indicada por el índice de imagen de referencia. La unidad de procesamiento de predicción 152 puede proporcionar un índice reldxL1 como entrada al proceso y puede establecer reldxL1 a la salida del proceso. RefldxL1 es un índice en la lista 1. La unidad de procesamiento de predicción 152 puede establecer inicialmente que reldxL1 sea igual a 0. El proceso para mover una imagen de referencia a una posición en la lista 1 indicada por el índice de imagen de referencia se describe a continuación. De lo contrario, si el elemento sintáctico "list\_modification\_idc" es igual a 3, la unidad de
- 35

procesamiento de predicción 152 detiene el proceso de modificación de la lista de imágenes de referencia para la lista 1.

5 **[176]** Como se ha mencionado anteriormente, la unidad de procesamiento de predicción 152 puede realizar un proceso para mover una imagen de referencia a una posición en la lista 0 o 1 indicada por el índice de imagen de referencia. Este proceso puede tomar el índice de imagen de referencia como entrada y generar un índice de imagen de referencia incrementado. El índice de imagen de referencia se puede denotar como `refIdxLX`, donde `X` es igual a 0 para el movimiento de una imagen de referencia en la lista 0 y `X` es igual a 1 para el movimiento de una imagen de referencia en la lista 1. Si el elemento sintáctico "`list_modification_idc`" es igual a 0 y la lista de imágenes de referencia actual es la lista 0, el descodificador de vídeo puede designar `RefPicSetStCurr0` como el conjunto de imágenes de referencia actual. De lo contrario, el elemento sintáctico "`list_modification_idc`" es igual a 0 y la lista de imágenes de referencia actual es la lista 1, la unidad de procesamiento de predicción 152 puede designar `RefPicStCurr1` como el conjunto de imágenes de referencia actual. Si el elemento sintáctico "`list_modification_idc`" es igual a 1 y la lista de imágenes de referencia actual es la lista 0, la unidad de procesamiento de predicción 152 puede designar `RefPicStCurr1` como el conjunto de imágenes de referencia actual. Si el elemento sintáctico "`list_modification_idc`" es igual a 1 y la lista de imágenes de referencia actual es la lista 1, la unidad de procesamiento de predicción 152 puede designar `RefPicStCurr0` como el conjunto de imágenes de referencia actual. Si el elemento sintáctico "`list_modification_idc`" es igual a 2, la unidad de procesamiento de predicción 152 puede designar `RefPicSetLtCurr` como el conjunto de imágenes de referencia actual. De acuerdo con las técnicas de esta divulgación, si el elemento sintáctico "`list_modification_idc`" es igual a 4, la unidad de procesamiento de predicción 152 puede designar `RefPicSetInterView` como el conjunto de imágenes de referencia actual. Además, de acuerdo con las técnicas de esta divulgación, si el elemento sintáctico "`id_modificación`" es igual a 5, la unidad de procesamiento de predicción 152 puede designar `RefPicSetVsrp` como el conjunto de imágenes de referencia actual. De forma alternativa, los valores particulares (por ejemplo, 4 y 5) del elemento sintáctico "`list_modification_idc`" pueden designar `RefPicSetInterViewFwd` o `RefPicSetInterViewBwd` como el conjunto de imágenes de referencia actual.

30 **[177]** Además, en el proceso para mover una imagen de referencia a una posición en la lista 0 o 1 indicada por el índice de imagen de referencia, la unidad de procesamiento de predicción 152 puede determinar un valor POC relevante, `pocLX`. En algunos ejemplos, la unidad de procesamiento de predicción 152 puede determinar `pocLX` como `PicOrderCnt( curRefPicSet[ ref_pic_set_idx ] )`, donde `PicOrderCnt( )` devuelve el valor POC de un componente de visualización y `curRefPicSet` es el conjunto de imágenes de referencia actual. El valor POC relevante puede ser el valor POC de la imagen de referencia en la lista de imágenes de referencia actual en la posición indicada por el elemento sintáctico "`ref_pic_set_idx`". Además, la unidad de procesamiento de predicción 152 puede determinar un índice de orden de visualización relevante, `vOldx`. En algunos ejemplos, la unidad de procesamiento de predicción 152 puede determinar `vOldx` como `viewOrderIdx( curRefPicSet[ ref_pic_set_idx ] )`, donde `viewOrderIdx( )` es una función que devuelve un índice de orden de visualización de un componente de visualización.

40 **[178]** Un VSRP puede tener los mismos valores POC que el componente de visualización actual y puede tener el mismo índice de orden de visualización que un componente de visualización desde el que se sintetiza originalmente el VSRP. En algunos ejemplos donde la unidad de procesamiento de predicción 152 sintetiza el VSRP desde múltiples componentes de visualización, el índice de orden de visualización de la imagen VSRP puede ser igual al índice de orden de visualización más pequeño de los componentes de visualización a partir de los cuales se sintetiza el VSRP. En otros ejemplos donde la unidad de procesamiento de predicción 152 sintetiza el VSRP desde múltiples componentes de visualización, el índice de orden de visualización de la imagen VSRP puede ser igual al índice de orden de visualización de cualquiera de los componentes de visualización a partir de los cuales se sintetiza el VSRP.

50 **[179]** La unidad de procesamiento de predicción 152 puede realizar un proceso descrito por el siguiente pseudocódigo para insertar una imagen que tiene un valor POC igual al valor POC relevante en la lista de imágenes de referencia actual en la posición indicada por `refIdxLX`.

```

for(cldx = num_ref_idx_1X_active_minus1 + 1; cldx > refIdxLX; cldx--)
55     RefPicListX[ cldx ] = RefPicListX[ cldx - 1]
    RefPicListX[ refIdxLX++ ] = pocLX
    nIdx = refIdxLX
60     for(cldx = refIdxLX; cldx <= num_ref_idx_1X_active_minus1 + 1; cldx++)
        if(PicOrderCnt(RefPicListX[ cldx ]) != pocLX || viewOrderIdx(RefPicListX[ cldx ]) != vOldx)
65         RefPicListX[ nIdx++ ] = RefPicListX[ cldx ]
    
```

**[180]** Cuando la unidad de procesamiento de predicción 152 realiza el proceso descrito por el pseudocódigo anterior, la unidad de procesamiento de predicción 152 puede desplazar las imágenes de referencia siguiendo la posición indicada por `refIdxLX` a posiciones posteriores en la lista de imágenes de referencia X, insertar una imagen de referencia que tenga el valor POC relevante (`pocLX`) en la lista de imágenes de referencia X, incrementar `refIdxLX` y eliminar un duplicado de la imagen de referencia insertada de la lista de imágenes de referencia X. En el pseudocódigo anterior, `cldx` es una variable de contador. Además, en el pseudocódigo anterior, `num_ref_idx_1X_active_minus1` es una variable igual al número de imágenes de referencia activas en la lista de imágenes de referencia actual, menos 1. `PicOrderCnt` es una función que devuelve un valor POC de una imagen de referencia. En el pseudocódigo anterior, la longitud de la lista de imágenes de referencia actual puede ser temporalmente un elemento más largo que la longitud de la versión final de la lista de imágenes de referencia actual. Por consiguiente, después de que la unidad de procesamiento de predicción 152 realiza la operación descrita por el pseudocódigo anterior, la unidad de procesamiento de predicción 152 puede retener solo los elementos 0 a `num_ref_idx_1X_active_minus1` en la versión final de la lista de imágenes de referencia actual.

**[181]** En el ejemplo anterior de proceso de modificación de lista de imágenes de referencia de movimiento, la unidad de procesamiento de predicción 152 determina un índice de orden de visualización relevante, `vOldx`, y usa el `vOldx` para determinar una posición dentro de una lista de imágenes de referencia en la que insertar una imagen de referencia. En otro ejemplo, la unidad de procesamiento de predicción 152 puede realizar un proceso de modificación de lista de imágenes de referencia similar al del ejemplo anterior, excepto que la unidad de procesamiento de predicción 152 determina y usa un identificador de visualización relevante, `viewId`, para identificar una imagen de referencia entre visualizaciones o VSRP en lugar de un índice de orden de visualización. Por ejemplo, la unidad de procesamiento de predicción 152 puede determinar el identificador de visualización relevante como `ViewId(curRefPicSet[ ref_pic_set_idx ])`, donde `ViewId` es una función que devuelve un identificador de visualización de un componente de visualización y `curRefPicSet` es el conjunto de imágenes de referencia actual. En este ejemplo, la unidad de procesamiento de predicción 152 puede realizar un procedimiento descrito por el siguiente pseudocódigo para insertar una imagen de referencia en una lista de imágenes de referencia.

```

for( cldx = num_ref_idx_1X_active_minus1 + 1; cldx > refIdxLX; cldx-- )
    RefPicListX[ cldx ] = RefPicListX[ cldx - 1 ]

    RefPicListX[ refIdxLX++ ] = pocLX

nIdx = refIdxLX

for( cldx = refIdxLX; cldx <= num_ref_idx_1X_active_minus1 + 1; cldx++ )

    if( PicOrderCnt( RefPicListX[ cldx ] ) != pocLX || viewIdx( RefPicListX[ cldx ] ) != viewId )

        RefPicListX[ nIdx++ ] = RefPicListX[ cldx ]

```

**[182]** En el análisis anterior, se describen ejemplos basados en la identificación de imágenes usando POC e índice de orden de visualización. Sin embargo, las mismas técnicas también pueden aplicarse en función de identificar imágenes usando POC e ID de visualización.

**[183]** Ahora se hace una referencia continua al ejemplo de la FIG. 5. La unidad de reconstrucción 158 puede utilizar los bloques de píxeles residuales asociados con las TU de una CU y los bloques de píxeles predictivos de las PU de la CU, es decir, datos de intra-predicción o datos de inter-predicción, según corresponda, para reconstruir el bloque de píxeles de la CU. En particular, la unidad de reconstrucción 158 puede agregar muestras de los bloques de píxeles residuales a las muestras correspondientes de los bloques de píxeles predictivos para reconstruir el bloque de píxeles de la CU.

**[184]** La unidad de filtro 159 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo asociados con el bloque de píxeles de la CU. El descodificador de vídeo 30 puede almacenar el bloque de píxeles de la CU en la memoria intermedia de imágenes descodificadas 160. La memoria intermedia de imágenes descodificadas 160 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intra-predicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en los bloques de píxeles en la memoria intermedia de imágenes descodificadas 160, operaciones de intra-predicción o de inter-predicción en las PU de otras CU.

**[185]** La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de operación 200 realizada por el codificador de vídeo 20 como parte de un proceso para codificar datos de vídeo, de acuerdo con una o más técnicas de esta divulgación. El diagrama de flujo de la FIG. 6 y los diagramas de flujo de las siguientes figuras se proporcionan como ejemplos. En otros ejemplos, las técnicas de esta divulgación se pueden implementar usando más, menos, o

diferentes etapas que las mostradas en el ejemplo de la FIG. 6 y las siguientes figuras.

**[186]** En el ejemplo de la FIG. 6, el codificador de vídeo 20 genera, basándose en un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso, una lista de imágenes de referencia para el componente de visualización actual, con el conjunto de imágenes de referencia que incluye un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones (202). Además, el codificador de vídeo 20 puede codificar el componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia (204). El codificador de vídeo 20 puede generar un flujo de bits que incluye elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual (206).

**[187]** La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de operación 230, realizada por el descodificador de vídeo 30 como parte de un proceso para descodificar datos de vídeo codificados, de acuerdo con una o más técnicas de esta divulgación. En el ejemplo de la FIG. 7, el descodificador de vídeo 30 puede analizar, desde un flujo de bits, elementos sintácticos que indican un conjunto de imágenes de referencia de un componente de visualización actual de una unidad de acceso, incluyendo el conjunto de imágenes de referencia un conjunto de imágenes de referencia entre visualizaciones que incluye una pluralidad de componentes de visualización que pertenecen a la unidad de acceso y que están asociados con diferentes visualizaciones (232). Además, el descodificador de vídeo 30 puede generar, basándose en el conjunto de imágenes de referencia, una lista de imágenes de referencia para el componente de visualización actual (234). El descodificador de vídeo 30 puede descodificar el componente de visualización actual basándose al menos en parte en una o más imágenes de referencia en la lista de imágenes de referencia (236).

**[188]** En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse, como una o más instrucciones o código, en un medio legible por ordenador o transmitirse a través de este, 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.

**[189]** A modo de ejemplo, y no de manera limitativa, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión 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.

**[190]** 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), conjuntos lógicos programables sobre el terreno (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. 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 descodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

**[191]** Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o

aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). 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.

5

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

10

## REIVINDICACIONES

1. Un procedimiento para descodificación de vídeo de múltiples visualizaciones, comprendiendo el procedimiento:

5 generar (234) una lista de imágenes de referencia para un componente de visualización actual, donde  
 generar la lista de imágenes de referencia comprende incluir en la lista de imágenes de referencia cada  
 componente de visualización de un primer subconjunto de un conjunto de imágenes de referencia del  
 componente de visualización actual, seguido de cada componente de visualización en un conjunto de  
 10 imágenes de referencia entre visualizaciones del conjunto de imágenes de referencia del componente de  
 visualización actual, seguido de cada componente de visualización de un segundo subconjunto del conjunto  
 de imágenes de referencia del componente de visualización actual, y seguido de cada componente de  
 visualización de un tercer subconjunto del conjunto de imágenes de referencia del componente de  
 visualización actual, donde el primer subconjunto consiste en componentes de visualización de referencia a  
 15 corto plazo que son anteriores al componente de visualización actual tanto en orden de descodificación  
 como de salida, el segundo subconjunto consiste en componentes de visualización de referencia a corto  
 plazo que son anteriores al componente de visualización actual en orden de descodificación y que vienen a  
 continuación del componente de visualización actual en orden de salida, el tercer subconjunto consiste en  
 componentes de visualización de referencia a largo plazo que son anteriores al componente de  
 20 visualización actual en orden de descodificación, el conjunto de imágenes de referencia entre  
 visualizaciones consiste en componentes de visualización que no son de una visualización actual y en el  
 que los componentes de visualización del conjunto de imágenes de referencia entre visualizaciones son de  
 una misma unidad de acceso que los componentes de visualización actuales y en el que los componentes  
 de visualización del conjunto de imágenes de referencia entre visualizaciones son utilizables por el  
 componente de visualización actual para la referencia de predicción entre visualizaciones; y

25 descodificar (236) al menos una parte del componente de visualización actual basándose en una o más  
 imágenes de referencia en la lista de imágenes de referencia.

2. El procedimiento según la reivindicación 1, en el que el conjunto de imágenes de referencia incluye además un  
 30 cuarto subconjunto, un quinto subconjunto y un sexto subconjunto, comprendiendo el cuarto subconjunto  
 componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización  
 actual tanto en orden de descodificación como en orden de salida que se pueden utilizar en inter-predicción de  
 uno o más componentes de visualización que vienen a continuación del componente de visualización actual y  
 que no se utilizan en inter-predicción del componente de visualización actual, comprendiendo el quinto  
 35 subconjunto componentes de visualización de referencia a corto plazo que son anteriores al componente de  
 visualización actual en orden de descodificación y vienen a continuación del componente de visualización  
 actual en orden de salida que pueden utilizarse en inter-predicción de uno o más componentes de visualización  
 que vienen a continuación del componente de visualización actual y que no se utilizan en la inter-predicción del  
 componente de visualización actual, comprendiendo el sexto subconjunto componentes de visualización de  
 40 referencia a largo plazo que son anteriores al componente de visualización actual en orden de descodificación  
 que se pueden utilizar en la inter-predicción de uno o más componentes de visualización que vienen a  
 continuación del componente de visualización actual en orden de descodificación y que no se usan en inter-  
 predicción del componente de visualización actual.

3. El procedimiento según la reivindicación 1, en el que la generación de la lista de imágenes de referencia  
 45 comprende generar la lista de imágenes de referencia de manera que el primer subconjunto o el segundo  
 subconjunto ocurran primero en la lista de imágenes de referencia, seguido del conjunto de imágenes de  
 referencia entre visualizaciones y preferentemente en el que la generación de la lista de imágenes de  
 referencia comprende generar la lista de imágenes de referencia de manera que el primer subconjunto y el  
 50 segundo subconjunto aparezcan en la lista de imágenes de referencia, seguido del conjunto de imágenes de  
 referencia entre visualizaciones y preferentemente en el que la generación de la lista de imágenes de  
 referencia comprenda generar la lista de imágenes de referencia de modo que el primer, el segundo, y el tercer  
 subconjuntos ocurran en la lista de imágenes de referencia, seguido del conjunto de imágenes de referencia  
 entre visualizaciones y que además comprende preferentemente realizar un proceso de modificación de lista de  
 55 imágenes de referencia que cambia una posición en la lista de imágenes de referencia de un componente de  
 visualización de referencia entre visualizaciones y preferentemente en el que el conjunto de imágenes de  
 referencia entre visualizaciones se señala en un nivel de secuencia y es compartido por todos los componentes  
 de visualización en una visualización actual.

4. El procedimiento según la reivindicación 1, en el que:

la lista de imágenes de referencia es una primera lista de imágenes de referencia y el conjunto de imágenes  
 de referencia entre visualizaciones comprende un primer subconjunto de imágenes de referencia entre  
 65 visualizaciones y un segundo subconjunto de imágenes de referencia entre visualizaciones, incluyendo el  
 primer subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de  
 referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de

5 visualización menores que un identificador de visualización del componente de visualización actual, incluyendo el segundo subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización mayores que el identificador de visualización del componente de visualización actual;

10 generar la primera lista de imágenes de referencia comprende generar la primera lista de imágenes de referencia de manera que la primera lista de imágenes de referencia incluye el primer subconjunto de imágenes de referencia entre visualizaciones y no el segundo subconjunto de imágenes de referencia entre visualizaciones; y

15 el procedimiento comprende además generar una segunda lista de imágenes de referencia de manera que la segunda lista de imágenes de referencia incluye el segundo subconjunto de imágenes de referencia entre visualizaciones y no el primer subconjunto de imágenes de referencia entre visualizaciones y preferentemente en el que:

20 la lista de imágenes de referencia es una primera lista de imágenes de referencia y el conjunto de imágenes de referencia entre visualizaciones comprende un primer subconjunto de imágenes de referencia entre visualizaciones y un segundo subconjunto de imágenes de referencia entre visualizaciones, incluyendo el primer subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización menores que un identificador de visualización del componente de visualización actual, incluyendo el segundo subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización mayores que el identificador de visualización del componente de visualización actual;

30 generar la primera lista de imágenes de referencia comprende generar la primera lista de imágenes de referencia de manera que la primera lista de imágenes de referencia incluye cada componente de visualización de referencia entre visualizaciones del primer subconjunto de imágenes de referencia entre visualizaciones antes de cualquier componente de visualización de referencia entre visualizaciones del segundo subconjunto de imágenes de referencia entre visualizaciones; y el procedimiento comprende además generar una segunda lista de imágenes de referencia tal que la segunda lista de imágenes de referencia incluye cada componente de visualización de referencia entre visualizaciones del segundo subconjunto de imágenes de referencia entre visualizaciones antes de cualquier componente de visualización de referencia entre visualizaciones del primer subconjunto de imágenes de referencia entre visualizaciones y además preferentemente comprendiendo además el análisis, desde una cabecera de fragmento, de elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual de la unidad de acceso.

40 **5.** Un procedimiento para codificación de vídeo de múltiples visualizaciones, comprendiendo el procedimiento:

45 generar (202) una lista de imágenes de referencia para un componente de visualización actual, en el que generar la lista de imágenes de referencia comprende incluir en la lista de imágenes de referencia cada componente de visualización de un primer subconjunto de un conjunto de imágenes de referencia del componente de visualización actual, seguido de cada componente de visualización en un conjunto de imágenes de referencia entre visualizaciones del conjunto de imágenes de referencia del componente de visualización actual, seguido de cada componente de visualización de un segundo subconjunto del conjunto de imágenes de referencia del componente de visualización actual, y seguido de cada componente de visualización de un tercer subconjunto del conjunto de imágenes de referencia del componente de visualización actual, consistiendo el primer subconjunto en componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual tanto en orden de descodificación como en orden de salida, consistiendo el segundo subconjunto en componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual en orden de descodificación y que vienen a continuación del componente de visualización actual en orden de salida, consistiendo el tercer subconjunto en componentes de visualización de referencia a largo plazo que son anteriores al componente de visualización actual en orden de descodificación, consistiendo el conjunto de imágenes de referencia entre visualizaciones en componentes de visualización que no son de una visualización actual y en el que los componentes de visualización del conjunto de imágenes de referencia entre visualizaciones son de una misma unidad de acceso que los componentes de visualización actuales, y en el que los componentes de visualización del conjunto de imágenes de referencia entre visualizaciones son utilizables por el componente de visualización actual para la referencia de predicción entre visualizaciones;

65 codificar (204) el componente de visualización actual basándose al menos en parte en una o más imágenes de referencia en la lista de imágenes de referencia; y

generar (206) un flujo de bits que incluye elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual.

5     **6.** Un dispositivo (30) para descodificación de vídeo de múltiples visualizaciones que comprende:

10           medios para generar una lista de imágenes de referencia para un componente de visualización actual, en el que generar la lista de imágenes de referencia comprende incluir en la lista de imágenes de referencia cada componente de visualización de un primer subconjunto de un conjunto de imágenes de referencia del componente de visualización actual, seguido de cada componente de visualización en un conjunto de imágenes de referencia entre visualizaciones del conjunto de imágenes de referencia del componente de visualización actual, seguido de cada componente de visualización de un segundo subconjunto del conjunto de imágenes de referencia del componente de visualización actual, y seguido de cada componente de visualización de un tercer subconjunto del conjunto de imágenes de referencia del componente de visualización actual, consistiendo el primer subconjunto en componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual tanto en orden de descodificación como de salida, consistiendo el segundo subconjunto en componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual en orden de descodificación y que vienen a continuación del componente de visualización actual en orden de salida, consistiendo el tercer subconjunto en componentes de visualización de referencia a largo plazo que son anteriores al componente de visualización actual en orden de descodificación, consistiendo el conjunto de imágenes de referencia entre visualizaciones en componentes de visualización que no provienen de una visualización actual y en el que los componentes de visualización del conjunto de imágenes de referencia entre visualizaciones son de una misma unidad de acceso que los componentes de visualización actuales y en los que los componentes de visualización del conjunto de imágenes de referencia entre visualizaciones son utilizables por el componente de visualización actual para la referencia de predicción entre visualizaciones; y

25           medios para descodificar al menos una parte del componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia.

30     **7.** El dispositivo de la reivindicación 6, en el que el conjunto de imágenes de referencia incluye además un cuarto subconjunto, un quinto subconjunto y un sexto subconjunto, comprendiendo el cuarto subconjunto componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual tanto en orden de descodificación como en orden de salida que se pueden utilizar en inter-predicción de uno o más componentes de visualización que vienen a continuación del componente de visualización actual y que no se utilizan en inter-predicción del componente de visualización actual, comprendiendo el quinto subconjunto componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual en orden de descodificación y vienen a continuación del componente de visualización actual en orden de salida que son utilizables en la inter-predicción de uno o más componentes de visualización que vienen a continuación del componente de visualización actual y que no se utilizan en la inter-predicción del componente de visualización actual, comprendiendo el sexto subconjunto componentes de visualización de referencia a largo plazo que son anteriores al componente de visualización actual en orden de descodificación que se pueden utilizar en la inter-predicción de uno o más componentes de visualización que vienen a continuación del componente de visualización actual en orden de descodificación y que no se utilizan en la inter-predicción del componente de visualización actual.

45     **8.** El dispositivo según la reivindicación 6, en el que uno o más procesadores están configurados para generar la lista de imágenes de referencia de manera que el primer subconjunto o el segundo subconjunto se encuentren primero en la lista de imágenes de referencia, seguido del conjunto de imágenes de referencia entre visualizaciones y preferentemente en el que uno o más procesadores están configurados para generar la lista de imágenes de referencia de modo que el primer subconjunto y el segundo subconjunto ocurran en la lista de imágenes de referencia, seguido del conjunto de imágenes de referencia entre visualizaciones y preferentemente en el que uno o más procesadores estén configurados para generar la lista de imágenes de referencia tal que el primer, segundo y tercer subconjuntos ocurran en la lista de imágenes de referencia, seguidos del conjunto de imágenes de referencia entre visualizaciones y preferentemente en el que el uno o más procesadores están configurados para realizar un proceso de modificación de lista de imágenes de referencia que cambia una posición en la lista de imágenes de referencia de un componente de visualización de referencia entre visualizaciones y, en particular, en el que el conjunto de imágenes de referencia entre visualizaciones se señala en un nivel de secuencia y es compartido por todos los componentes de visualización en una visualización actual.

60     **9.** El dispositivo de la reivindicación 6, en el que:

65           la lista de imágenes de referencia es una primera lista de imágenes de referencia y el conjunto de imágenes de referencia entre visualizaciones comprende un primer subconjunto de imágenes de referencia entre visualizaciones y un segundo subconjunto de imágenes de referencia entre visualizaciones, incluyendo el

primer subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización menores que un identificador de visualización del componente de visualización actual, incluyendo el segundo subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización mayores que el identificador de visualización del componente de visualización actual;

el uno o más procesadores están configurados para:

generar la primera lista de imágenes de referencia de manera que la primera lista de imágenes de referencia incluya el primer subconjunto de imágenes de referencia entre visualizaciones y no el segundo subconjunto de imágenes de referencia entre visualizaciones; y

generar una segunda lista de imágenes de referencia de manera que la segunda lista de imágenes de referencia incluya el segundo subconjunto de imágenes de referencia entre visualizaciones y no el primer subconjunto de imágenes de referencia entre visualizaciones.

**10.** El dispositivo de la reivindicación 6, en el que:

la lista de imágenes de referencia es una primera lista de imágenes de referencia y el conjunto de imágenes de referencia entre visualizaciones comprende un primer subconjunto de imágenes de referencia entre visualizaciones y un segundo subconjunto de imágenes de referencia entre visualizaciones, incluyendo el primer subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización menores que un identificador de visualización del componente de visualización actual, incluyendo el segundo subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización mayores que el identificador de visualización del componente de visualización actual;

el uno o más procesadores están configurados para:

generar la primera lista de imágenes de referencia de manera que la primera lista de imágenes de referencia incluya cada componente de visualización de referencia entre visualizaciones del primer subconjunto de imágenes de referencia entre visualizaciones antes de cualquier componente de visualización de referencia entre visualizaciones del

segundo subconjunto de imágenes de referencia entre visualizaciones; y

generar una segunda lista de imágenes de referencia tal que la segunda lista de imágenes de referencia incluye cada componente de visualización de referencia entre visualizaciones del segundo subconjunto de imágenes de referencia entre visualizaciones antes de cualquier componente de visualización de referencia entre visualizaciones del primer subconjunto de imágenes de referencia entre visualizaciones y preferentemente en el que el uno o más procesadores están configurados para analizar, desde una cabecera de fragmento, elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual de la unidad de acceso.

**11.** Un dispositivo de codificación de vídeo de múltiples visualizaciones (20), que comprende:

medios para generar una lista de imágenes de referencia para un componente de visualización actual, en el que generar la lista de imágenes de referencia comprende incluir en la lista de imágenes de referencia cada componente de visualización de un primer subconjunto de un conjunto de imágenes de referencia del componente de visualización actual, seguido de cada componente de visualización en un conjunto de imágenes de referencia entre visualizaciones del conjunto de imágenes de referencia del componente de visualización actual, seguido de cada componente de visualización de un segundo subconjunto del conjunto de imágenes de referencia del componente de visualización actual, y seguido de cada componente de visualización de un tercer subconjunto del conjunto de imágenes de referencia del componente de visualización actual, consistiendo el primer subconjunto en componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual tanto en orden de descodificación como de salida, consistiendo el segundo subconjunto en componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual en orden de descodificación y que vienen a continuación del componente de visualización actual en orden de salida, consistiendo el tercer subconjunto en componentes de visualización de referencia a largo plazo que son anteriores al componente de visualización actual en orden de descodificación, consistiendo el conjunto de imágenes de referencia entre visualizaciones en componentes de visualización que no son de una visualización actual y en el que

los componentes de visualización del conjunto de imágenes de referencia entre visualizaciones son de una misma unidad de acceso que los componentes de visualización actuales y en el que los componentes de visualización del conjunto de imágenes de referencia entre visualizaciones son utilizables por el componente de visualización actual para la referencia de predicción entre visualizaciones;

5 medios para codificar el componente de visualización actual basándose en una o más imágenes de referencia en la lista de imágenes de referencia; y

10 medios para generar un flujo de bits que incluye elementos sintácticos que indican el conjunto de imágenes de referencia del componente de visualización actual.

12. El dispositivo de codificación de vídeo de la reivindicación 11, en el que el conjunto de imágenes de referencia incluye además un cuarto subconjunto, un quinto subconjunto y un sexto subconjunto, comprendiendo el cuarto subconjunto componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual tanto en orden de descodificación como en orden de salida que se pueden utilizar en inter-predicción de uno o más componentes de visualización que vienen a continuación del componente de visualización actual y que no se utilizan en la inter-predicción del componente de visualización actual, comprendiendo el quinto subconjunto componentes de visualización de referencia a corto plazo que son anteriores al componente de visualización actual en orden de descodificación y vienen a continuación del componente de visualización actual en orden de salida que pueden utilizarse en la inter-predicción de uno o más componentes de visualización que vienen a continuación del componente de visualización actual y que no se utilizan en inter-predicción del componente de visualización actual, comprendiendo el sexto subconjunto componentes de visualización de referencia a largo plazo que son anteriores al componente de visualización actual en orden de descodificación que se pueden utilizar en la inter-predicción de uno o más componentes de visualización que vienen a continuación del componente de visualización actual en orden de descodificación y que no se usan en la inter-predicción del componente de visualización actual.

13. El dispositivo de codificación de vídeo de la reivindicación 11, en el que el uno o más procesadores están configurados para generar la lista de imágenes de referencia de modo que el primer subconjunto o el segundo subconjunto aparece primero en la lista de imágenes de referencia, seguido del conjunto de imágenes de referencia entre visualizaciones y preferiblemente en el que el uno o más procesadores están configurados para generar la lista de imágenes de referencia de manera que el primer subconjunto y el segundo subconjunto aparezcan en la lista de imágenes de referencia, seguido del conjunto de imágenes de referencia entre visualizaciones y más preferentemente en el que el uno o más procesadores están configurados para generar la lista de imágenes de referencia de modo que el primer, segundo y tercer subconjuntos aparezcan en la lista de imágenes de referencia, seguido del conjunto de imágenes de referencia entre visualizaciones y además preferentemente en el que uno o más procesadores están configurados para realizar un proceso de modificación de la lista de imágenes de referencia que cambia una posición en la lista de imágenes de referencia de un componente de visualización de referencia entre visualizaciones y, en particular, en el que el conjunto de imágenes de referencia entre visualizaciones es señalado en un nivel de secuencia y compartido por todos los componentes de visualización en una visualización actual y preferentemente en el que:

la lista de imágenes de referencia es una primera lista de imágenes de referencia y el conjunto de imágenes de referencia entre visualizaciones comprende un primer subconjunto de imágenes de referencia entre visualizaciones y un segundo subconjunto de imágenes de referencia entre visualizaciones, incluyendo el primer subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización menores que un identificador de visualización del componente de visualización actual, incluyendo el segundo subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización mayores que el identificador de visualización del componente de visualización actual; y

el uno o más procesadores están configurados para:

generar la primera lista de imágenes de referencia de manera que la primera lista de imágenes de referencia incluya el primer subconjunto de imágenes de referencia entre visualizaciones y no el segundo subconjunto de imágenes de referencia entre visualizaciones; y

generar una segunda lista de imágenes de referencia de manera que la segunda lista de imágenes de referencia incluya el segundo subconjunto de imágenes de referencia entre visualizaciones y no el primer subconjunto de imágenes de referencia entre visualizaciones.

14. El dispositivo de codificación de vídeo según la reivindicación 11, en el que:

la lista de imágenes de referencia es una primera lista de imágenes de referencia y el conjunto de imágenes

de referencia entre visualizaciones comprende un primer subconjunto de imágenes de referencia entre visualizaciones y un segundo subconjunto de imágenes de referencia entre visualizaciones, incluyendo el primer subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización menores que un identificador de visualización del componente de visualización actual, incluyendo el segundo subconjunto de imágenes de referencia entre visualizaciones componentes de visualización de referencia del conjunto de imágenes de referencia entre visualizaciones que tienen identificadores de visualización mayores que el identificador de visualización del componente de visualización actual; y

el uno o más procesadores están configurados para:

generar la primera lista de imágenes de referencia de manera que la primera lista de imágenes de referencia incluya cada componente de visualización de referencia entre visualizaciones del primer subconjunto de imágenes de referencia entre visualizaciones antes de cualquier componente de visualización de referencia entre visualizaciones del segundo subconjunto de imágenes de referencia entre visualizaciones; y

generar una segunda lista de imágenes de referencia tal que la segunda lista de imágenes de referencia incluya cada componente de visualización de referencia entre visualizaciones del segundo subconjunto de imágenes de referencia entre visualizaciones antes de cualquier componente de visualización de referencia entre visualizaciones del primer subconjunto de imágenes de referencia entre visualizaciones y preferentemente en el que el flujo de bits incluya una cabecera de fragmento que incluya los elementos sintácticos que indican el conjunto de imágenes de referencia.

15. Un medio de almacenamiento legible por ordenador que tiene instrucciones ejecutables por ordenador almacenadas en el mismo que, cuando son ejecutadas por uno o más procesadores de un dispositivo de codificación de vídeo de múltiples visualizaciones, configuran el dispositivo de codificación de vídeo para ejecutar el procedimiento de una cualquiera de las reivindicaciones 1 a 5.

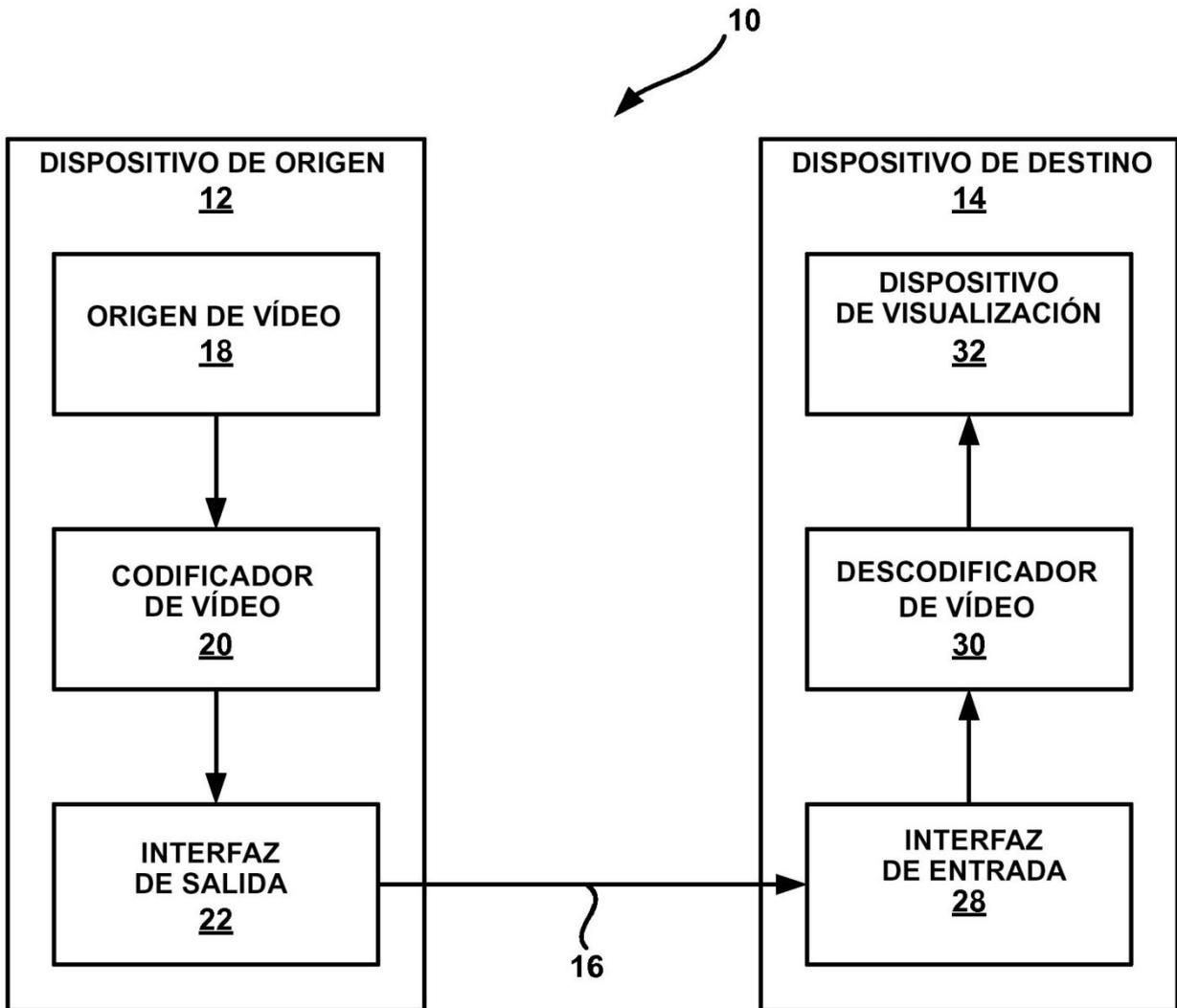


FIG. 1

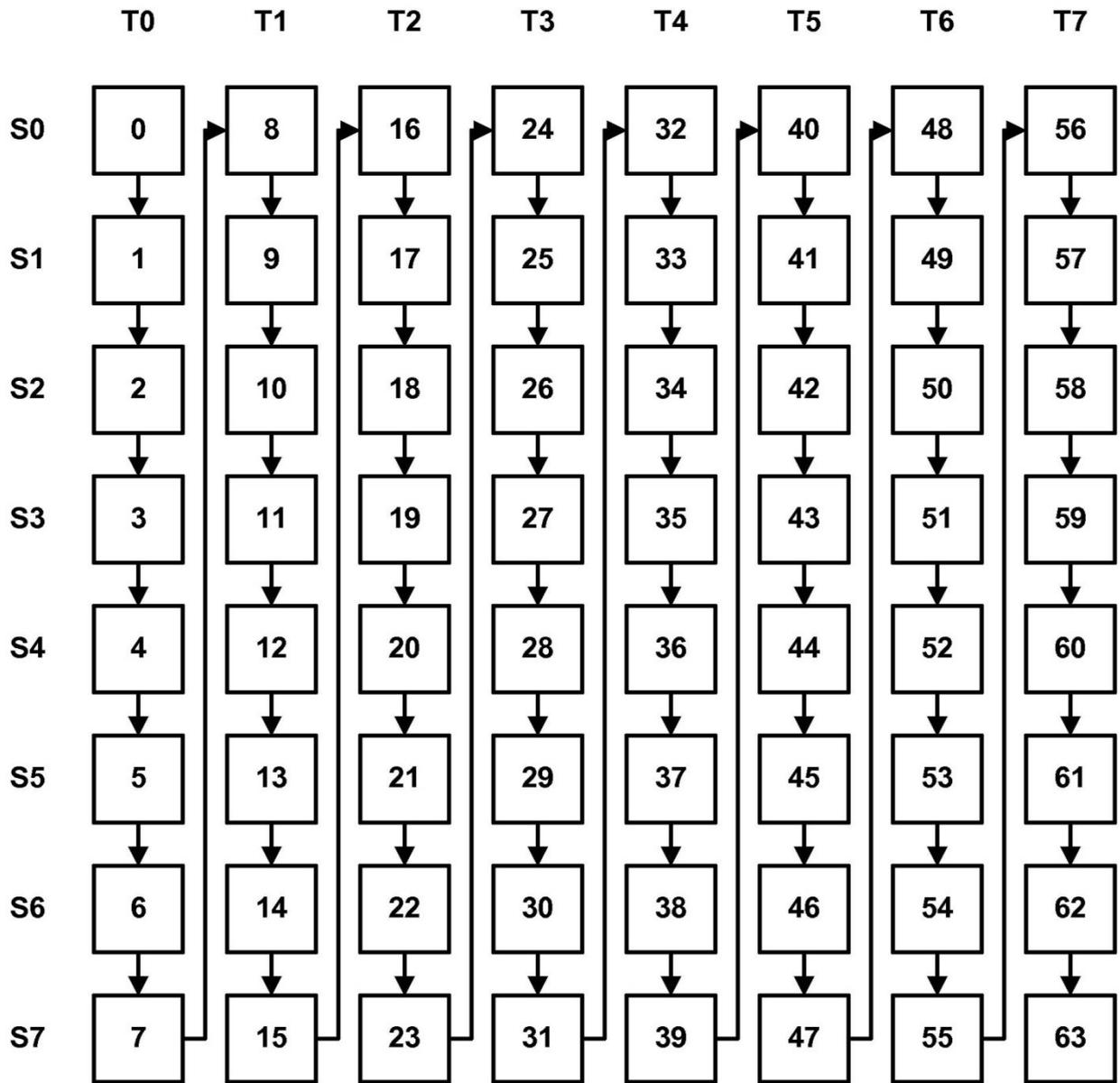


FIG. 2

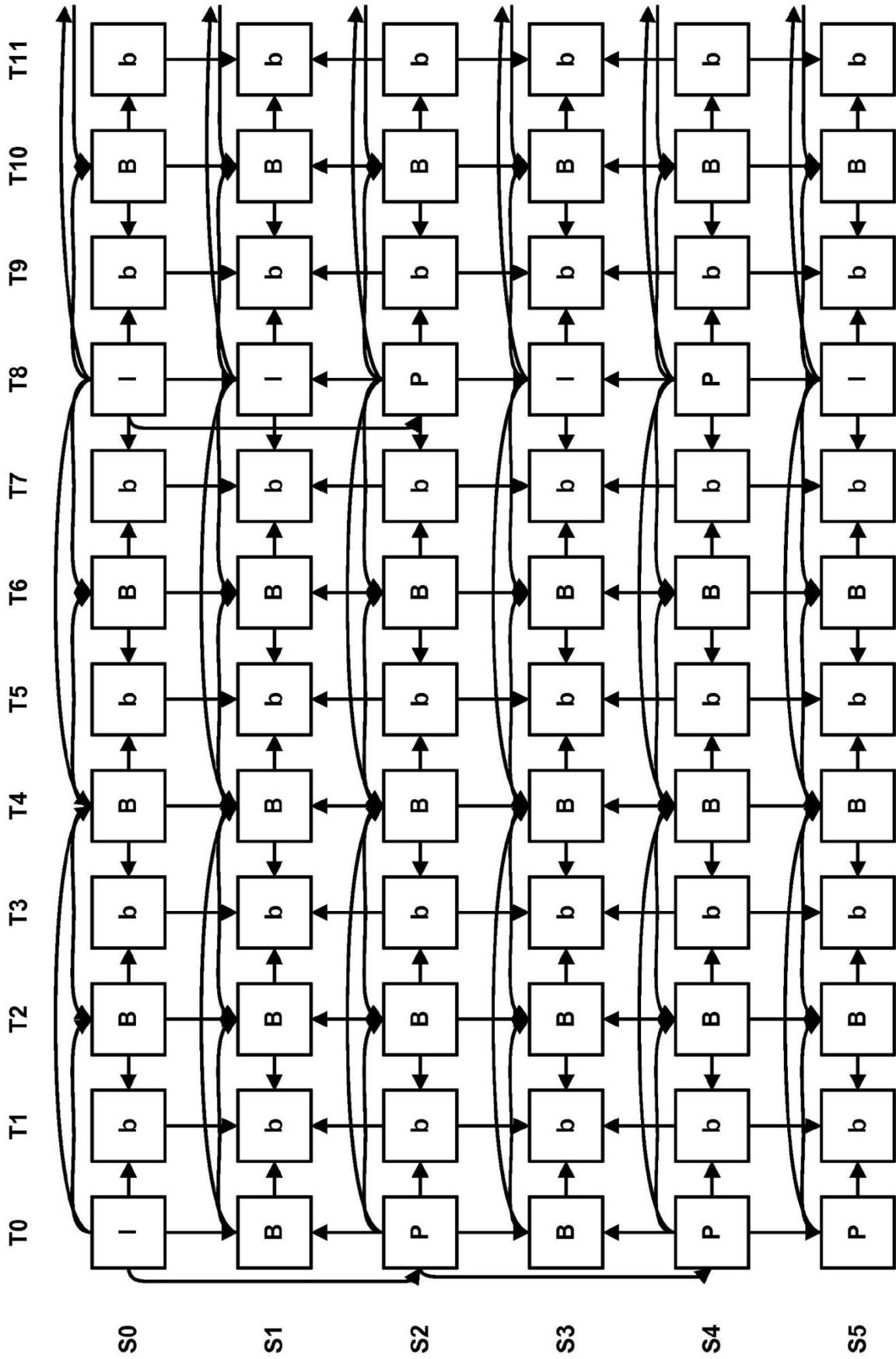


FIG. 3

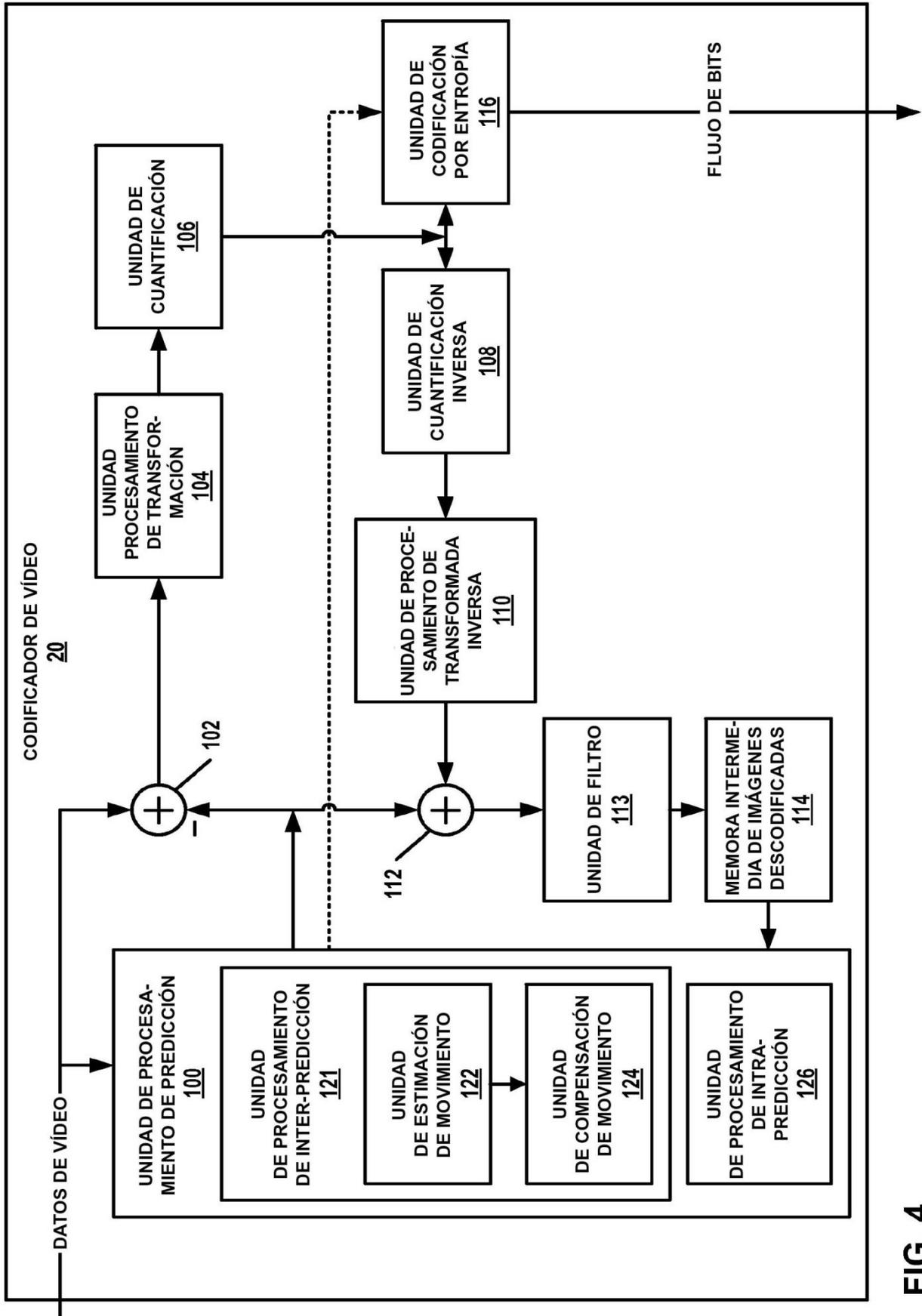


FIG. 4

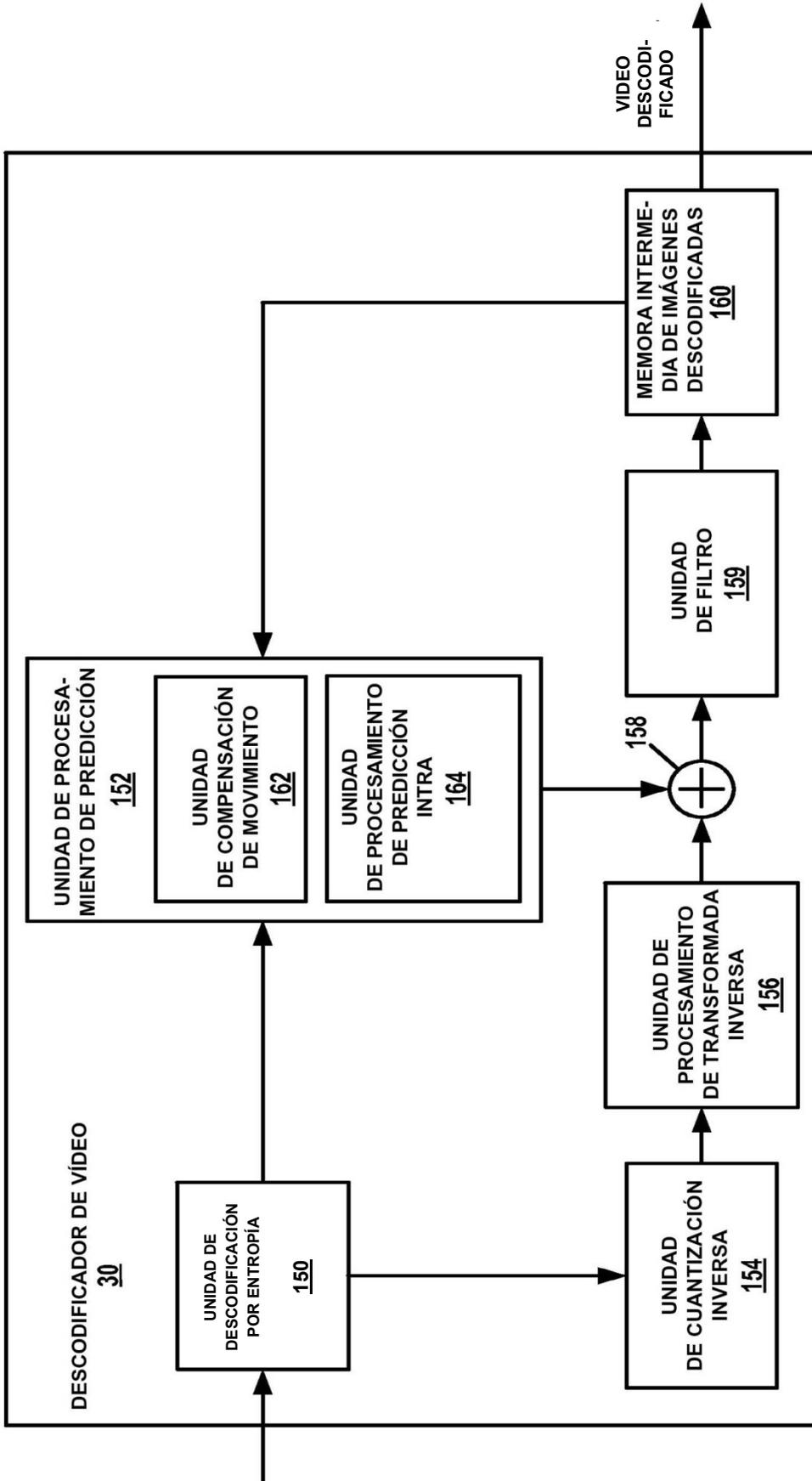


FIG. 5

200

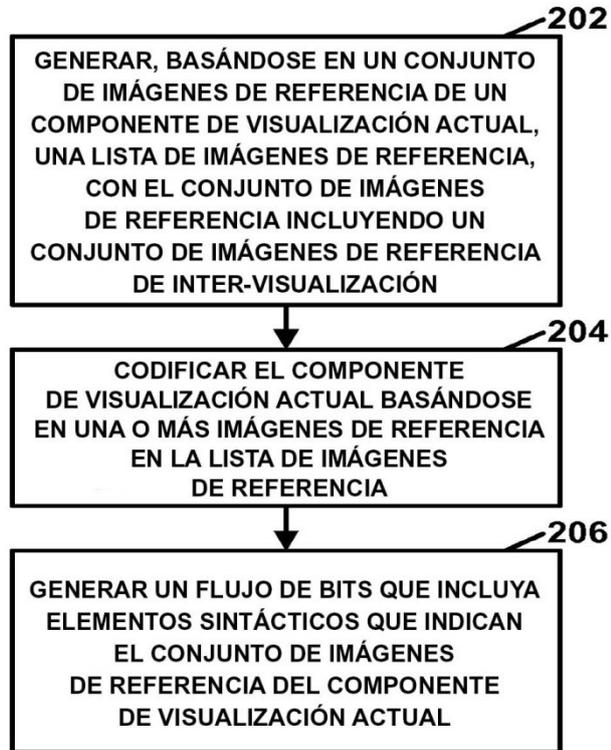


FIG. 6

230

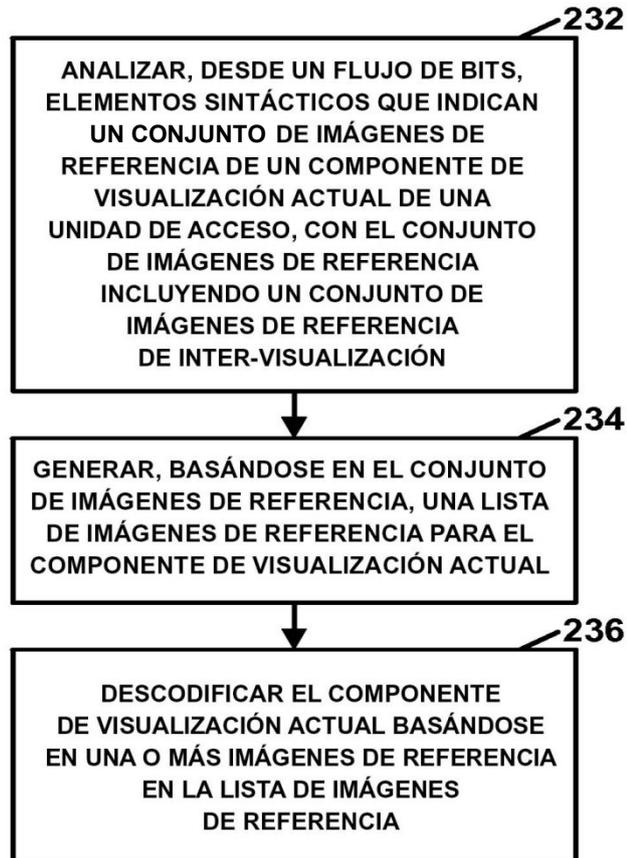


FIG. 7