

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 605 392**

51 Int. Cl.:

H04N 19/50 (2014.01)

H04N 19/70 (2014.01)

H04N 19/31 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **16.12.2013 PCT/US2013/075421**

87 Fecha y número de publicación internacional: **03.07.2014 WO14105485**

96 Fecha de presentación y número de la solicitud europea: **16.12.2013 E 13817810 (8)**

97 Fecha y número de publicación de la concesión europea: **24.08.2016 EP 2939427**

54 Título: **Refinamiento progresivo con soporte de ajustabilidad temporal a escala en la codificación de vídeo**

30 Prioridad:

30.12.2012 US 201261747347 P
27.09.2013 US 201314040014

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
14.03.2017

73 Titular/es:

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

72 Inventor/es:

WANG, YE-KUI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 605 392 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

Refinamiento progresivo con soporte de ajustabilidad temporal a escala en la codificación de vídeo

5 CAMPO TÉCNICO

Esta divulgación se refiere a la codificación de vídeo y, más en particular, a técnicas para el refinamiento progresivo de datos de vídeo.

10 ANTECEDENTES

Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados teléfonos "inteligentes", dispositivos de videoconferencia, dispositivos de transmisión 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), 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, decodificar y/o almacenar información de vídeo digital más eficazmente, implementando tales técnicas de compresión de vídeo.

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 (por ejemplo, una trama de vídeo o una porción de una trama de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques arbolados, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intra-codificado (I) de una imagen son codificados usando la predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo de un fragmento inter-codificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen, o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.

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 a codificar y el bloque predictivo. Un bloque inter-codificado se codifica según 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 según una modalidad de intra-codificación y a datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, los cuales pueden cuantificarse posteriormente. Los coeficientes de transformación cuantificados, inicialmente dispuestos en una formación bidimensional, pueden escanearse con el fin de producir un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación por entropía para lograr más compresión.

45 RESUMEN

En general, esta divulgación describe técnicas para el refinamiento progresivo de datos de vídeo en flujos de bits de vídeo con ajustabilidad temporal a escala.

En un primer ejemplo, un procedimiento de decodificación de datos de vídeo incluye la recepción, en un mensaje de información de mejora suplementaria (SEI), de un recuento de orden de imágenes delta, POC, valor que indica una diferencia entre un valor del POC de una imagen que precede inmediatamente a una pluralidad de imágenes en un segmento de refinamiento progresivo en un flujo de bits de vídeo codificado, y un valor del POC de una última imagen entre una pluralidad de imágenes en un segmento de refinamiento progresivo en un flujo de bits codificado y, en base al valor de delta de POC recibido, la decodificación de al menos algunas de las imágenes en el segmento de refinamiento progresivo, según una operación de refinamiento progresivo, donde la operación de refinamiento progresivo indica un refinamiento con respecto a la calidad de una imagen.

En un segundo ejemplo, un procedimiento de codificación de datos de vídeo incluye la codificación, según una operación de refinamiento progresivo, de al menos algunas imágenes entre una pluralidad de imágenes en un segmento de refinamiento progresivo, en el que el segmento de refinamiento progresivo indica un refinamiento con respecto a la calidad de una imagen, la generación de un recuento de orden de imágenes delta, POC, valor que indica una diferencia entre un valor del POC de una imagen que precede inmediatamente al segmento de refinamiento progresivo en un flujo de bits codificado, y el valor del POC de una última imagen del segmento de refinamiento progresivo en un flujo de bits codificado, y que incluye, en un mensaje de información de mejora

suplementaria, un mensaje de SEI para ser señalado en el flujo de bits de vídeo codificado, el valor de diferencia de POC generado.

5 En un tercer ejemplo, un medio de almacenamiento legible por ordenador tiene almacenadas en el mismo instrucciones que, al ser ejecutadas, hacen que un procesador programable de un dispositivo informático lleve a cabo el procedimiento del primer ejemplo.

10 En un cuarto ejemplo, un dispositivo para decodificar datos de vídeo incluye medios para determinar, basándose, al menos en parte, en un mensaje de SEI De información de mejora suplementaria, un recuento de orden de imágenes delta, POC, valor que indica una diferencia entre un valor del POC de una imagen que precede inmediatamente a una pluralidad de imágenes en un segmento de refinamiento progresivo en un flujo de bits codificado, y un valor del POC de una última imagen entre una pluralidad de imágenes en un segmento de refinamiento progresivo en un flujo de bits codificado, y medios para la decodificación, en base a la información determinada, de al menos algunas de las imágenes en el segmento de refinamiento progresivo, según una operación de refinamiento progresivo, en el que la operación de refinamiento progresivo indica un refinamiento con respecto a la calidad de una imagen.

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

20 BREVE DESCRIPCIÓN DE LOS DIBUJOS

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo que puede implementar una o más técnicas descritas en esta divulgación.

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

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

30 La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de segmento de refinamiento progresivo y una imagen de base correspondiente.

35 La FIG. 5 es un diagrama de flujo que ilustra un ejemplo de proceso que un decodificador de vídeo y/o componentes del mismo pueden realizar, para decodificar datos de vídeo codificados con refinamiento progresivo, según uno o más aspectos de esta divulgación.

40 La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de proceso que el codificador de vídeo y/o los componentes de vídeo del mismo pueden realizar para codificar datos de vídeo con refinamiento progresivo, según uno o más aspectos de esta divulgación.

DESCRIPCIÓN DETALLADA

45 En general, las técnicas de esta divulgación están orientadas a la codificación de datos de vídeo usando el refinamiento progresivo, soportando al mismo tiempo la ajustabilidad temporal a escala de los datos de vídeo codificados. De acuerdo con diversos ejemplos de esta divulgación, un dispositivo de codificación de vídeo puede utilizar los mensajes proporcionados por el mecanismo de información de mejora suplementaria (SEI), soportados por ambos estándares AVC y HEVC, para permitir el refinamiento progresivo soportando al mismo tiempo la ajustabilidad temporal a escala. De esta manera, las técnicas de esta divulgación pueden ofrecer la ventaja potencial de permitir que los dispositivos de codificación de vídeo hagan uso de las infraestructuras existentes de hardware, software y comunicación, mejorando al mismo tiempo la codificación basada en refinamiento progresivo para soportar la ajustabilidad temporal a escala.

55 Un borrador reciente de la norma HEVC, denominado "Borrador 10 de trabajo de la HEVC" o "WD10", se describe en el documento JCTVC-H1003, de Bross et al., titulado "High Efficiency Video Coding (HEVC) Text Specification Draft 10 (for FDIS & Last Call)" ["Memoria descriptiva textual de la Codificación de Vídeo de Alta Eficacia (HEVC), Borrador 10 (para Norma Internacional de Borrador Final y Última Llamada)"], Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, décimo segunda conferencia: Ginebra, Suiza, 14 al 23 de enero de 2013, que, a partir del 6 de junio de 2013 puede descargarse desde:

60 http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip. Todo el contenido del documento WD10 se incorpora por la presente por referencia.

65 El estándar AVC (UIT-T) H.264 se describe en la Recomendación UIT-T H.264, Codificación de Vídeo Avanzada para los servicios audiovisuales genéricos, por el Grupo de Estudio de la UIT-T, y con fecha de marzo de 2005, que se puede denominar en este documento el estándar H.264 o la especificación H.264, o el estándar o especificación

H.264 / AVC. El Equipo de Vídeo Conjunto (JVT) continúa trabajando en extensiones para H.264 / MPEG-4 AVC.

Otro reciente Borrador de Trabajo (BT) de la HEVC, y denominado HEVC WD9 en adelante, está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCT-VC-K1003-v8.zip. Todo el contenido del documento HEVC WD9 (BROSS et al, "Especificación textual de la codificación de vídeo de alta eficacia (HEVC), borrador 9," Documento JCTVC-K1003_v7, 11.ª Reunión: Shanghai, CN, 10 al 19 de octubre de 2012, 290 págs.) se incorpora en el presente documento por referencia.

Ambos estándares AVC y HEVC soportan la codificación de datos de vídeo usando el refinamiento progresivo. El refinamiento progresivo puede permitir a un dispositivo codificar un conjunto de imágenes, como una secuencia o serie de imágenes dispuestas en orden de decodificación. Tal secuencia de imágenes se denomina aquí un "segmento de refinamiento progresivo". El segmento de refinamiento progresivo puede incluir dos o más imágenes codificadas, cada una de las cuales representa una versión refinada de una imagen particular (por ejemplo, una "imagen de base"). La imagen puede refinarse en términos de calidad u otras características tales como el formato espacial. En general, ninguna imagen codificada del segmento de refinamiento progresivo puede predecirse a partir de la imagen de base utilizando la predicción basada en movimiento. Un dispositivo de codificación de vídeo puede determinar los límites del segmento de refinamiento progresivo usando determinados mensajes de SEI, tales como los mensajes de SEI "inicio de segmento de refinamiento progresivo" y "fin de segmento de refinamiento progresivo".

Además, ambos estándares AVC y HEVC soportan la ajustabilidad temporal a escala de flujos de bits de vídeo. La ajustabilidad temporal a escala puede permitir a un dispositivo de codificación de vídeo determinar que un subconjunto de datos de vídeo codificados se puede extraer de un flujo de bits completo de datos de vídeo codificados. Un subconjunto de ese tipo de datos de vídeo codificados (por ejemplo, imágenes codificadas), extraídos de un flujo de bits completo según la ajustabilidad temporal a escala, puede denominarse un "subconjunto temporal." A su vez, la ajustabilidad temporal a escala, según el soporte prestado por los estándares AVC y HEVC, puede permitir a un dispositivo de codificación de vídeo determinar múltiples subconjuntos temporales a partir de un flujo de bits completo, de manera que los diferentes subconjuntos temporales incluyan números variables de imágenes codificadas. Un subconjunto temporal inferior, o "más aproximado", puede incluir un número menor de imágenes codificadas a partir del flujo de bits completo, y puede representar una velocidad inferior de imágenes o una velocidad inferior de tramas. A la inversa, un subconjunto temporal superior, o "más fino", puede incluir un mayor número de imágenes codificadas a partir del flujo de bits completo, y puede representar una velocidad superior de imágenes o una velocidad superior de tramas.

Los dispositivos de codificación de vídeo configurados para aplicar las implementaciones existentes de la codificación basada en el refinamiento progresivo de flujos de bits ajustables temporalmente a escala pueden encontrarse con, o presentar, una o más inexactitudes potenciales con respecto al segmento de refinamiento progresivo. Por ejemplo, según las implementaciones existentes del refinamiento progresivo, un elemento sintáctico incluido en el mensaje de SEI de inicio de segmento de refinamiento progresivo puede indicar un cierto número de imágenes codificadas consecutivas que forman el segmento de refinamiento progresivo.

En consecuencia, en casos en los que se señala un subconjunto temporal, el número de imágenes codificadas consecutivas en el segmento de refinamiento progresivo, según lo indicado por el elemento sintáctico del mensaje de SEI, puede ser inexacto. Más específicamente, como el subconjunto temporal representa un menor número de imágenes codificadas que el flujo de bits completo, una o más imágenes codificadas del segmento original de refinamiento progresivo pueden estar ausentes. Sin embargo, el elemento sintáctico que indica el número de imágenes en el segmento de refinamiento progresivo no se puede actualizar dinámicamente para reflejar el número reducido de imágenes codificadas en el segmento correspondiente de refinamiento progresivo del subconjunto temporal extraído. En consecuencia, puede haber un desajuste entre el número de imágenes codificadas consecutivas, que forman el segmento de refinamiento progresivo para el flujo de bits, y el número de imágenes codificadas consecutivas en un segmento correspondiente de un subconjunto temporal de ese flujo de bits.

Para aliviar o eliminar potencialmente tales inexactitudes de la codificación basada en el refinamiento progresivo, con respecto a flujos de bits temporalmente ajustables a escala, las técnicas de esta divulgación pueden permitir a un dispositivo de codificación de vídeo determinar los límites de un segmento de refinamiento progresivo utilizando la información que indica un valor de recuento de orden de imágenes (POC) de la última imagen codificada del segmento de refinamiento progresivo. Por ejemplo, cada imagen codificada del segmento de refinamiento progresivo puede estar asociada a un valor del POC estático. Como resultado, la información (por ejemplo, un elemento sintáctico) puede proporcionar una identificación precisa de la última imagen codificada del segmento, independientemente de si el segmento se incluye o no en un subconjunto temporal de un flujo de bits completo. En una implementación de las técnicas, la información puede indicar una diferencia entre los respectivos valores del POC (o "delta_POC") de la imagen de base y de la última imagen codificada del segmento de refinamiento progresivo. Según otra implementación de las técnicas, la información puede indicar una diferencia entre los respectivos bits menos significativos (LSB) de los respectivos valores del POC de la imagen de base y de la última imagen codificada del segmento de refinamiento progresivo.

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo

que puede utilizar las técnicas descritas en esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados, a decodificar en un momento posterior por parte de un dispositivo de destino 14. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, equipos telefónicos portátiles tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, un dispositivo de flujos de transmisión de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a decodificar, mediante un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificado desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir al dispositivo de origen 12 transmitir datos de vídeo codificado directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden ser modulados según una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitidos al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

Como alternativa, los datos codificados pueden ser emitidos desde la interfaz de salida 22 a un dispositivo de almacenamiento 31. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 31 mediante una interfaz de entrada. El dispositivo de almacenamiento 31 puede incluir cualquiera entre una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco, discos Blu-ray, discos DVD, discos CD-ROM, memoria flash, memoria volátil o no volátil, u otros medios adecuados cualesquiera de almacenamiento digital, para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento 31 puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda retener los datos de vídeo codificado, generados por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados procedentes del dispositivo de almacenamiento 31, mediante flujos de transmisión o descarga. El servidor de ficheros puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Entre los ejemplos de servidores de ficheros se incluyen un servidor de internet (por ejemplo, para un sitio web), un servidor del FTP, dispositivos de almacenamiento conectados a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificado almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificado desde el dispositivo de almacenamiento 31 puede ser una transmisión por flujo, una transmisión de descarga o una combinación de ambas.

Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de cualquiera entre una diversidad de aplicaciones de multimedia, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por flujo, por ejemplo, mediante Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para soportar la transmisión de vídeo unidireccional o bidireccional, para soportar aplicaciones tales como los flujos de transmisión de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo de origen 12, el origen de vídeo 18 puede incluir un origen tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos de ordenador para generar datos de gráficos de ordenador como el vídeo de origen, o una combinación de tales orígenes. Como un ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videotelefonos. Sin embargo, las técnicas descritas en esta divulgación pueden aplicarse a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas.

El vídeo capturado, pre-capturado o generado por ordenador puede ser codificado por el codificador de vídeo 20. Los datos de vídeo codificados pueden ser transmitidos directamente al dispositivo de destino 14 mediante la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados también (o como alternativa)

pueden almacenarse en el dispositivo de almacenamiento 31 para un acceso posterior por parte del dispositivo de destino 14 u otros dispositivos, para decodificación y/o reproducción.

El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados por el enlace 16. Los datos de vídeo codificado, comunicados por el enlace 16, o proporcionados en el dispositivo de almacenamiento 31, pueden incluir una diversidad de elementos sintácticos generados por el codificador de vídeo 20, para su uso por un decodificador de vídeo, tal como el decodificador de vídeo 30, en la decodificación de los datos de vídeo. Tales elementos sintácticos pueden incluirse con los datos de vídeo codificados, transmitidos en un medio de comunicación, almacenarse en un medio de almacenamiento o almacenarse en un servidor de ficheros.

El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también configurarse para mantener interfaces con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 muestra los datos de vídeo decodificado a un usuario, y puede comprender cualquiera entre una diversidad 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.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar según una norma de compresión de vídeo, tal como la norma de Codificación de Vídeo de Alta Eficacia (HEVC), actualmente en desarrollo, y pueden ser conformes al Modelo de Prueba de la HEVC (HM). Como alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar según otras normas patentadas o industriales, tales como la norma ITU-T H.264, denominada alternativamente MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), o extensiones de tales normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de compresión de vídeo incluyen la MPEG-2 y la ITU-T H.263.

Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar integrados en un codificador y decodificador de audio, y pueden incluir unidades adecuadas de multiplexado y demultiplexado, u otro hardware y software, para llevar a cabo la codificación, tanto de audio como de vídeo, en un flujo de datos común o en flujos de datos diferentes. Si procede, en algunos ejemplos, las unidades de multiplexado y demultiplexado pueden ajustarse al protocolo de multiplexado ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse como cualquiera entre una diversidad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables en el terreno (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas son implementadas parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio no transitorio adecuado, legible por ordenador, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden incluirse en uno o más codificadores o decodificadores, donde cualquiera de los mismos puede estar integrado como parte de un codificador/decodificador combinado (CÓDEC) en un dispositivo respectivo.

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

En general, el modelo de funcionamiento del HM describe que una trama o imagen de vídeo puede dividirse en una secuencia de bloques arbolados, o unidades de codificación (LCU) más grandes, que incluyen muestras tanto de luma como de croma. Un bloque arbolado tiene un fin similar al de un macro-bloque de la norma H.264. Un fragmento incluye un cierto número de bloques arbolados consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque arbolado puede dividirse en unidades de codificación (CU) según un árbol cuádruple. Por ejemplo, un bloque arbolado, como un nodo raíz del árbol cuádruple, puede dividirse en cuatro nodos secundarios, y cada nodo secundario puede a su vez ser un nodo principal y dividirse en otros cuatro nodos secundarios. Un nodo secundario final, no dividido, como un nodo de hoja del árbol cuádruple, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces en que puede dividirse un bloque arbolado, y también pueden definir un tamaño mínimo de los nodos de codificación.

Una CU puede incluir un bloque de codificación de luma y dos bloques de codificación de croma. La CU puede tener

unidades de predicción (PU) y unidades de transformación (TU) asociadas. Cada una de las PU puede incluir un bloque de predicción de luma y dos bloques de predicción de croma, y cada una de las TU puede incluir un bloque de transformación de luma y dos bloques de transformación de croma. Cada uno de los bloques de codificación puede dividirse en uno o más bloques de predicción que comprenden bloques para muestras a las que se aplica la misma predicción. Cada uno de los bloques de codificación también puede dividirse en uno o más bloques de transformación que comprenden bloques de muestra, sobre los que se aplica la misma transformación.

Un tamaño de la CU corresponde, en general, a un tamaño del nodo de codificación, y es habitualmente de forma cuadrada. El tamaño de la CU puede variar desde 8 x 8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64 x 64 píxeles, o más. Cada CU puede definir una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división del bloque de codificación en uno o más bloques de predicción. Las modalidades de división pueden diferir entre si la CU está codificada en modalidad de omisión o directa, codificada en modalidad de intra-predicción o codificada en modalidad de inter-predicción. Los bloques de predicción pueden dividirse para que tengan forma cuadrada o no cuadrada. Los datos sintácticos incluidos en una CU también pueden describir, por ejemplo, la división del bloque de codificación en uno o más bloques de transformación, según un árbol cuádruple. Los bloques de transformación pueden dividirse para que tengan forma cuadrada o no cuadrada.

La norma HEVC admite transformaciones según las TU, que pueden ser distintas para distintas CU. Las TU son habitualmente dimensionadas en base al tamaño de las PU dentro de una CU dada, definida para una LCU dividida, aunque esto puede no ser siempre el caso. Las TU tienen habitualmente el mismo tamaño, o un tamaño más pequeño, que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas usando una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos de hoja del RQT pueden representar los TUs. Los valores de diferencias de píxeles asociados a las TU pueden ser transformados para producir coeficientes de transformación, que pueden cuantificarse.

En general, una PU incluye datos relacionados con el proceso de predicción. Por ejemplo, cuando la PU está codificada en la modalidad de inter-predicción, la PU puede incluir datos que describen una modalidad de intra-predicción para la PU. Como otro ejemplo, cuando la PU está codificada en la inter-modalidad, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de píxeles de un cuarto o precisión de píxeles de un octavo), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, la Lista 0, la Lista 1 o la Lista C) para el vector de movimiento.

En general, se usa una TU para los procesos de transformación y cuantificación. Una CU dada con una o más PU también puede incluir una o más TU. Después de la predicción, el codificador de vídeo 20 puede calcular los valores residuales a partir del bloque de vídeo identificado por el nodo de codificación, según la PU. El nodo de codificación se actualiza a continuación, para hacer referencia a los valores residuales en lugar del bloque de vídeo original. Los valores residuales comprenden valores de diferencias de píxeles que pueden transformarse en coeficientes de transformación, cuantificarse y escanearse usando las transformaciones y otra información de transformación especificada en las TU, para producir coeficientes de transformación serializados para la codificación por entropía. El nodo de codificación puede, una vez más, ser actualizado para referirse a estos coeficientes de transformación serializados. Esta divulgación usa habitualmente el término "bloque de vídeo" para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar el término "bloque de vídeo" para referirse a un bloque arbolado, es decir, una LCU, o una CU, que incluye un nodo de codificación y las PU y las TU.

Una secuencia de vídeo incluye normalmente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende generalmente una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos, en una cabecera del GOP, en una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen un cierto número de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen una modalidad de codificación para el fragmento respectivo. Un codificador de vídeo 20 actúa habitualmente sobre bloques de vídeo dentro de los fragmentos de vídeo individuales, con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño, según una norma de codificación especificada.

Como ejemplo, el HM soporta la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU específica sea $2N \times 2N$, el HM soporta la intra-predicción en tamaños de PU de $2N \times 2N$ o $N \times N$, y la inter-predicción en tamaños de PU simétrica de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también soporta la división asimétrica para la inter-predicción en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida entre el 25 % y el 75 %. La parte de la CU correspondiente a la división del 25 % está indicada por una "n" seguida por una indicación de "Arriba", "Abajo", "Izquierda" o "Derecha". Así, por ejemplo, " $2N \times nU$ " se refiere a una CU de tamaño $2N \times 2N$ que está dividida horizontalmente, con una PU de tamaño $2N \times 0,5N$ encima y una PU de tamaño $2N \times 1,5N$ debajo.

- En esta divulgación, "NxN" y "N por N" pueden usarse de manera intercambiable para hacer referencia a las dimensiones de píxeles de un bloque vídeo, en lo que respecta a dimensiones verticales y horizontales, por ejemplo, 16x16 píxeles o 16 por 16 píxeles. En general, un bloque de tamaño 16 x 16 tendrá 16 píxeles en la dirección vertical ($y = 16$) y 16 píxeles en la dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño NxN tiene generalmente N píxeles en la dirección vertical y N píxeles en la dirección horizontal, donde N representa un valor entero no negativo. Los píxeles en un bloque pueden estar dispuestos en filas y columnas. Además, los bloques no necesitan tener necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender N x M píxeles, donde M no es necesariamente igual a N.
- Tras la codificación intra-predictiva o inter-predictiva, usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales a los que se aplican las transformaciones especificadas por las TU de la CU. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las CU. El codificador de vídeo 20 puede formar los datos residuales para la CU, y a continuación transformar los datos residuales para producir coeficientes de transformación.
- Tras cualquier transformación para producir coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación se refiere generalmente a un proceso en el que los coeficientes de transformación se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos de, o todos, los coeficientes. Por ejemplo, un valor de n bits puede redondearse por lo bajo a un valor de m bits durante la cuantificación, donde n es mayor que m .
- En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de escaneado predefinida para escanear los coeficientes de transformación cuantificados, para producir un vector serializado que pueda ser codificado por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneado adaptativo. Después de escanear los coeficientes de transformación cuantificados, para formar un vector unidimensional, el codificador de vídeo 20 puede codificar por entropía el vector unidimensional, por ejemplo, según la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto y de base sintáctica (SBAC), la codificación por Entropía por División de Intervalos de Probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede codificar por entropía elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el decodificador de vídeo 30, en la decodificación de los datos de vídeo.
- Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo contextual a un símbolo a transmitir. El contexto puede referirse, por ejemplo, a si los valores adyacentes del símbolo son distintos de cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. Las palabras de código en la VLC pueden ser construidas de modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de VLC puede lograr un poco de ahorro con respecto, por ejemplo, a usar palabras de código de igual longitud para cada símbolo a transmitir. La determinación de la probabilidad puede basarse en un contexto asignado al símbolo.
- El codificador de vídeo 20 y/o el decodificador de vídeo 30, pueden implementar las técnicas de esta divulgación para codificar datos de vídeo, utilizando el refinamiento progresivo, soportando al mismo tiempo flujos de bits ajustables temporalmente a escala. El codificador de vídeo 20 puede estar configurado, o ser accionable de otra manera, para codificar una serie o secuencia de imágenes para formar un segmento de refinamiento progresivo. A su vez, el codificador de vídeo 20 puede indicar el segmento de refinamiento progresivo, como parte de un flujo de bits de vídeo codificado, al decodificador de vídeo 30.
- El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden identificar el segmento de refinamiento progresivo como una secuencia de imágenes que sigue inmediatamente a una imagen de base en el orden de decodificación. Para identificar el segmento de refinamiento progresivo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar que un mensaje de SEI de inicio de segmento de refinamiento progresivo sigue inmediatamente a la imagen de base. Además, ninguna imagen del segmento de refinamiento progresivo se predice a partir de la imagen de base, directa o indirectamente, usando la información de movimiento. En algunos ejemplos, cada imagen del segmento de refinamiento progresivo representa un refinamiento de calidad incremental definido sobre la imagen anterior de la secuencia de refinamiento progresivo.
- Por ejemplo, la primera imagen del segmento de refinamiento progresivo puede representar un refinamiento de calidad de un incremento definido en comparación con la imagen de base. A su vez, la segunda imagen del segmento de refinamiento progresivo puede representar un refinamiento de calidad de un incremento definido, con respecto a la primera imagen del segmento de refinamiento progresivo. En varios ejemplos, los incrementos definidos pueden seguir siendo los mismos en todo el segmento de refinamiento progresivo, o pueden variar en diferentes transiciones de imágenes dentro del segmento de refinamiento progresivo. De esta manera, el segmento de refinamiento progresivo, tomado en su totalidad, puede representar una serie coherente de refinamientos de calidad sobre la imagen de base.

5 El codificador de vídeo 20 puede codificar la imagen de base y, a su vez, puede codificar el segmento de refinamiento progresivo aplicando el refinamiento gradual de la calidad a la imagen codificada que precede inmediatamente a cada imagen del segmento de refinamiento progresivo. Además, el codificador de vídeo 20 puede
 10 señalar los límites del segmento de refinamiento progresivo usando uno o más mensajes de información de mejora suplementaria (SEI). Por ejemplo, el codificador de vídeo 20 puede señalar el límite inicial del segmento de refinamiento progresivo usando un mensaje de SEI de "inicio de segmento de refinamiento progresivo", por ejemplo, inmediatamente antes de la primera imagen en el segmento de refinamiento progresivo. De manera similar, el codificador de vídeo 20 puede señalar el límite final del segmento de refinamiento progresivo usando un mensaje de SEI de "fin de segmento de refinamiento progresivo", por ejemplo, inmediatamente después de la última imagen en el segmento de refinamiento progresivo. Una visión general de los mensajes de SEI, según el soporte en un borrador de trabajo de la HEVC (por ejemplo, "WD9"), se ilustra en la Tabla 1 a continuación.

15 **Tabla 1. Visión general de Mensajes de SEI**

Mensaje de SEI	Propósito
Período de almacenamiento temporal	Retrasos iniciales para la operación del decodificador de referencia hipotética (HRD)
Temporización de imágenes	Tiempo de salida de la imagen y tiempo de eliminación de imagen / sub-imagen para la operación del HRD, así como información relacionada con la estructura de la imagen
Rectángulo de escaneado panorámico	Exhibición con una relación de aspecto de imagen (PAR) diferente a la PAR de las imágenes de salida
Carga útil de relleno	Ajuste de la velocidad de bits para satisfacer limitaciones específicas
Datos de usuario registrados - Datos de usuario no registrados	Mensajes de SEI a especificar por entidades externas
Punto de recuperación	Información adicional para acceso aleatorio puro. Actualización gradual de decodificación.
información de escena	Información sobre cambios de escena y transiciones
Instantánea a pantalla completa	Indicación para etiquetar la imagen decodificada asociada como una instantánea de imagen fija del contenido de vídeo
Segmento de refinamiento progresivo	Indica que ciertas imágenes consecutivas representan un refinamiento progresivo de la calidad de una imagen, en lugar de una escena en movimiento
Características de textura de película	Permite a los decodificadores sintetizar la textura de la película
Preferencia de exhibición de filtro desbloqueador	Recomienda si las imágenes exhibidas deberían someterse o no al proceso del filtro desbloqueador de bucle interno
Sugerencia de pos-filtro	Proporciona coeficientes de pos-filtro sugeridos o información de correlación para el diseño de pos-filtro
Información de correlación de tonos	Re-correlación con otro espacio de colores distinto al usado o adoptado en la codificación
Disposición de empaquetamiento de tramas	Empaquetamiento de vídeo estereoscópico en un flujo de bits de HEVC
Orientación de la pantalla	Especifica el vuelco y/o la rotación que deberían aplicarse a las imágenes de salida cuando se exhiben
Estructura de descripción de imágenes	Describe la estructura temporal y de inter-predicción del flujo de bits
Información no válida de imagen decodificada	Suma de comprobación de la imagen decodificada, que puede ser utilizada para la detección de errores
Conjuntos de parámetros activos	Proporciona información sobre VPS, SPS, etc., activos
Información de unidad de decodificación	Tiempo de eliminación de sub-imagen para la operación del HRD, así como el índice de unidad de decodificación
Índice cero del nivel temporal	Proporciona valores de índice cero de nivel temporal
Almacenamiento ajustable a escala	Proporciona un mecanismo para almacenar mensajes de SEI para la asociación a diferentes puntos de operación y capas
Información de actualización de regiones	Proporciona información sobre regiones actualizadas y no actualizadas para la actualización de decodificación gradual

La sintaxis y la semántica para el mensaje de SEI de inicio de segmento de refinamiento progresivo, según el soporte de HEVC WD9, se ilustra en la tabla de sintaxis 1 a continuación.

5 **Tabla de sintaxis 1**

inicio de segmento de refinamiento progresivo (tamañoDeCargaÚtil) {	Descriptor
Identificador_de_refinamiento_progresivo	ue(v)
núm_de_etapas_de_refinamiento_menos1	ue(v)
}	

10 Como se ha descrito, el codificador de vídeo 20 puede codificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para especificar el comienzo del segmento de refinamiento progresivo. En consecuencia, el decodificador de vídeo 30 puede, al decodificar un flujo recibido de bits de vídeo codificado, decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo y determinar el comienzo del segmento codificado de refinamiento progresivo, por ejemplo, para identificar la primera imagen en el segmento codificado de refinamiento progresivo. En base a la señalización del codificador de vídeo 20 del conjunto de imágenes consecutivas en el segmento de refinamiento progresivo, según el orden de decodificación, el decodificador de vídeo 30 puede decodificar el segmento de refinamiento progresivo en el orden señalado. Además, en algunos ejemplos, el codificador de vídeo 20 puede generar y señalar el segmento de refinamiento progresivo como un conjunto de imágenes codificadas consecutivas en orden de salida. En estos ejemplos, el decodificador de vídeo 30 también puede decodificar el segmento de refinamiento progresivo en el orden de salida.

El elemento sintáctico "identificador_de_refinamiento_progresivo" incluido en el mensaje de SEI de inicio de segmento de refinamiento progresivo puede identificar la operación de refinamiento progresivo que define el refinamiento gradual de la calidad entre imágenes consecutivas del segmento de refinamiento progresivo. Por ejemplo, el codificador de vídeo 20 puede establecer el valor del elemento sintáctico `identificador_de_refinamiento_progresivo` dentro de una gama numérica que se inicia en 0 y termina en $(2^{32}-2)$, ambos valores inclusive. Más específicamente, el codificador de vídeo 20 puede establecer el valor del elemento sintáctico `identificador_de_refinamiento_progresivo` para indicar una operación particular de refinamiento progresivo que define el refinamiento incremental de la calidad entre imágenes consecutivas del segmento de refinamiento progresivo.

A su vez, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo, para obtener el valor señalado del elemento sintáctico `identificador_de_refinamiento_progresivo`. Además, el decodificador de vídeo 30 puede utilizar el valor del elemento sintáctico `identificador_de_refinamiento_progresivo` para determinar la operación de refinamiento progresivo a aplicar a la imagen de base codificada y señalizada, para decodificar la primera imagen codificada del segmento de refinamiento progresivo.

Del mismo modo, el decodificador de vídeo 30 puede aplicar la operación de refinamiento progresivo indicada por el elemento sintáctico `identificador_de_refinamiento_progresivo` a cada imagen del segmento de refinamiento progresivo, a excepción de la última imagen del segmento de refinamiento progresivo. En otras palabras, el decodificador de vídeo 30 puede aplicar iterativamente la operación identificada por el valor del elemento sintáctico `identificador_de_refinamiento_progresivo`, comenzando con la primera imagen del segmento de refinamiento progresivo, y terminando en la penúltima imagen del segmento de refinamiento progresivo. Más específicamente, mediante la aplicación de la operación de refinamiento identificada por el valor del elemento sintáctico `identificador_de_refinamiento_progresivo` a la imagen de base codificada y a cada una, desde la primera a la penúltima imagen codificada, del segmento de refinamiento progresivo, el decodificador de vídeo 30 puede decodificar todo el segmento de refinamiento progresivo.

Además, en la codificación del mensaje de SEI de inicio de segmento de refinamiento progresivo, el codificador de vídeo 20 puede establecer el valor del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` para indicar el número de imágenes codificadas incluidas en el conjunto etiquetado de imágenes codificadas que componen el segmento de refinamiento progresivo. Más específicamente, en casos en que el codificador de vídeo 20 establece el valor del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` en un valor no nulo, el número de imágenes codificadas incluidas en el segmento de refinamiento progresivo puede ser igual al valor del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1`, más 1.

A su vez, el decodificador de vídeo 30 puede utilizar el valor del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` para determinar que un cierto número de aplicaciones iterativas de la operación de refinamiento progresivo son necesarias para decodificar todo el segmento de refinamiento progresivo. Más específicamente, el decodificador de vídeo 30 puede aplicar la operación de refinamiento progresivo indicada por el valor del elemento sintáctico `identificador_de_refinamiento_progresivo` a la imagen de base, para decodificar

la primera imagen codificada del segmento de refinamiento progresivo. Además, el decodificador de vídeo 30 puede aplicar iterativamente la misma operación de refinamiento progresivo a una serie de imágenes codificadas que siguen inmediatamente a la imagen de base en el orden de decodificación.

5 Más específicamente, el decodificador de vídeo 30 puede aplicar iterativamente la operación de refinamiento progresivo identificada a un número de imágenes codificadas consecutivas, que es igual al valor del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1`. Debido a que el elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` denota un valor que es uno menos que el número total de imágenes codificadas incluidas en el segmento de refinamiento progresivo, el decodificador de vídeo 30 puede aplicar iterativamente la operación de refinamiento progresivo a cada imagen codificada, a excepción de la última imagen codificada, del segmento de refinamiento progresivo. En otras palabras, el decodificador de vídeo 30 puede aplicar los valores indicados por los elementos sintácticos `identificador_de_refinamiento_progresivo` y `núm_de_etapas_de_refinamiento_menos1` para aplicar la misma operación de refinamiento progresivo a la imagen de base codificada, y a una serie de imágenes codificadas, que comienza en la primera entrada del segmento de refinamiento progresivo y que termina en la penúltima entrada del segmento de refinamiento progresivo.

Además, el codificador de vídeo 20 puede codificar y señalar un mensaje de SEI de "fin de segmento de refinamiento progresivo", para especificar o indicar de otro modo el final del segmento de refinamiento progresivo. En algunos ejemplos, el codificador de vídeo 20 puede señalar el mensaje de SEI de fin de segmento de refinamiento progresivo, conjuntamente con, o después de, la última imagen codificada del segmento de refinamiento progresivo. La sintaxis y la semántica para el mensaje de SEI de fin de segmento de refinamiento progresivo, según el soporte disponible en HEVC WD9, se ilustran en la tabla de sintaxis 2 a continuación.

Tabla de sintaxis 2

25

<code>fin_de_segmento_de_refinamiento_progresivo</code> (tamañoDeCargaÚtil) {	Descriptor
<code> Identificador_de_refinamiento_progresivo</code>	<code>ue(v)</code>
}	

Como se muestra en la anterior tabla de sintaxis 2, el codificador de vídeo 20 puede generar el mensaje de SEI de fin de segmento de refinamiento progresivo para incluir un elemento sintáctico `identificador_de_refinamiento_progresivo`. Más específicamente, el codificador de vídeo 20 puede establecer el valor del elemento sintáctico `identificador_de_refinamiento_progresivo` en el mismo valor que el del elemento sintáctico `identificador_de_refinamiento_progresivo` especificado en el correspondiente mensaje de SEI de inicio de segmento de refinamiento progresivo para un segmento actual de refinamiento progresivo. Como se ha descrito, el codificador de vídeo 20 puede establecer el valor del elemento sintáctico `identificador_de_refinamiento_progresivo` dentro de una gama numérica que comienza en 0 y termina en $(2^{32}-2)$, ambos valores inclusive.

A su vez, el decodificador de vídeo 30 puede decodificar el mensaje codificado de SEI de fin de segmento de refinamiento progresivo, recibido en el flujo de bits de vídeo codificado, para determinar el límite final del segmento de refinamiento progresivo. En particular, el decodificador de vídeo 30 puede determinar, en base a la detección del mensaje de SEI de fin de segmento de refinamiento progresivo, dejar de aplicar la operación de refinamiento progresivo para decodificar las imágenes codificadas posteriores, señalizadas en el flujo de bits de vídeo codificado. Como ejemplo, el decodificador de vídeo 30 puede determinar que la primera imagen codificada después del segmento de refinamiento progresivo se predice usando información de movimiento proveniente de una imagen codificada anterior, señalizada en el flujo de bits de vídeo codificado.

Además, según el documento HEVC WD9, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden soportar la ajustabilidad temporal a escala de flujos de bits de vídeo codificado. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden soportar diferentes velocidades de imagen (o "velocidades de tramas") proporcionadas por los diferentes flujos de bits de vídeo codificado. Por ejemplo, para soportar una velocidad de imagen temporal inferior a un flujo completo de bits de vídeo codificado, el codificador de vídeo 20 puede señalar un subconjunto del flujo completo de bits de vídeo codificado. En un ejemplo específico, el codificador de vídeo 20 puede señalar un subconjunto de las imágenes codificadas incluidas en el flujo completo de bits de vídeo codificado. En otras palabras, el subconjunto señalado de imágenes codificadas puede incluir al menos una imagen codificada menos, en comparación con el flujo completo de bits de vídeo codificado. Un subconjunto de imágenes codificadas, señalizadas por el codificador de vídeo 20 para soportar una velocidad de imagen inferior, se denomina en este documento un "subconjunto temporal", o un "sub-flujo de bits".

El codificador de vídeo 20 puede señalar diferentes subconjuntos temporales, según diferentes velocidades de imagen proporcionadas mediante la ajustabilidad temporal a escala de flujos de bits de vídeo codificado. En un ejemplo, el codificador de vídeo 20 puede soportar una baja velocidad de imagen, señalizando un primer subconjunto temporal del flujo completo de bits de vídeo codificado. Según este ejemplo, el codificador de vídeo 20 puede soportar una velocidad de imagen intermedia, señalizando un segundo subconjunto temporal, que incluye al menos una imagen codificada más que el primer subconjunto temporal, pero al menos una imagen codificada menos que el flujo completo de bits de vídeo codificado. En este ejemplo, el codificador de vídeo 20 puede señalar una

velocidad posible de imagen máxima señalizando la totalidad (por ejemplo, el conjunto completo de imágenes codificadas) del flujo completo de bits de vídeo codificado.

5 Sin embargo, en los casos en que el codificador de vídeo 20 señala un segmento de refinamiento progresivo como parte de un subconjunto temporal según el documento HEVC WD9, el codificador de vídeo 20 puede no estar configurado para actualizar dinámicamente el valor del `núm_de_etapas_de_refinamiento_menos1` incluido en el mensaje de SEI de inicio de segmento de refinamiento progresivo. Como resultado, el codificador de vídeo 20 puede señalar potencialmente un recuento inexacto de imágenes codificadas incluidas en el segmento de refinamiento progresivo temporalmente ajustado a escala. Más específicamente, en la generación de un subconjunto temporal o sub-flujo de bits a partir del flujo completo de bits de vídeo codificado, el codificador de vídeo 20 puede eliminar una o más imágenes codificadas del flujo completo de bits, de tal manera que el número de imágenes que corresponden a un segmento de refinamiento progresivo en el subconjunto temporal sea inferior al provisto para el conjunto completo de imágenes.

15 En algunos escenarios, el codificador de vídeo 20 puede eliminar una o más imágenes codificadas que se incluyen en el segmento de refinamiento progresivo. En un escenario en el que el codificador de vídeo 20 elimina una o más imágenes codificadas del segmento de refinamiento progresivo para ajustar temporalmente a escala el flujo de bits completo, el elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` puede representar un valor incorrecto. Más específicamente, el valor del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` puede indicar el número de imágenes codificadas en el segmento original de refinamiento progresivo, según lo representado en el flujo de bits completo, disminuido en uno. Sin embargo, en este escenario, el segmento de refinamiento progresivo temporalmente ajustado a escala puede incluir un número de imágenes codificadas menor que el segmento original de refinamiento progresivo del flujo de bits completo. En consecuencia, el codificador de vídeo puede generar y señalar el elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` con un valor incorrecto, con respecto al
25 segmento de refinamiento progresivo temporalmente ajustado a escala.

A su vez, el decodificador de vídeo 30 puede decodificar el mensaje codificado de SEI de inicio de segmento de refinamiento progresivo, y obtener así un valor incorrecto indicado por el elemento sintáctico `núm_de_etapas_de_refinamiento_menos1`, con respecto al segmento de refinamiento progresivo temporalmente ajustado a escala. Como resultado, el decodificador de vídeo 30 puede aplicar la operación de refinamiento progresivo identificada por el `identificador_de_refinamiento_progresivo` a un número de imágenes codificadas consecutivas que supere el número de imágenes codificadas en el segmento de refinamiento progresivo temporalmente ajustado a escala. En algunos de tales ejemplos, el decodificador de vídeo 30 puede decodificar una imagen codificada usando solo la operación de refinamiento progresivo, donde, de hecho, la imagen codificada no representa un refinamiento de la calidad de la imagen de base, o representa un refinamiento de la calidad además de otras modificaciones, tales como la predicción basada en el movimiento.

40 Para mitigar o eliminar las inexactitudes potenciales descritas anteriormente con respecto a los segmentos de refinamiento progresivo ajustados temporalmente a escala, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden implementar una o más técnicas de esta divulgación. En algunas implementaciones de las técnicas descritas en el presente documento, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden utilizar un valor de recuento de orden de imágenes (POC) de la última entrada del segmento de refinamiento progresivo para identificar los límites del segmento de refinamiento progresivo en un flujo de bits de vídeo codificado, independientemente de si el flujo de bits de vídeo codificado es un subconjunto temporal o un flujo de bits completo. Como el valor del POC de una imagen codificada puede ser un valor estático, tal vez no sea necesario el decodificador de vídeo 30 para obtener el valor del POC del límite final del segmento de refinamiento progresivo en base a una serie de imágenes codificadas que están señalizadas.

50 En cambio, según algunas implementaciones de las técnicas descritas en el presente documento, el decodificador de vídeo 30 puede detectar el límite final de una secuencia de refinamiento progresivo, en base al valor del POC de la última imagen codificada del segmento, en el momento de la decodificación de la última imagen codificada del segmento. En diversos ejemplos, la información codificada por el codificador de vídeo 20 puede permitir al decodificador de vídeo 30 detectar la última imagen codificada del segmento, mediante la generación y señalización de información que indica el valor del POC de la última imagen del segmento en un flujo de bits de vídeo codificado.

55 En diversos ejemplos, el codificador de vídeo 20 puede implementar una o más técnicas de esta divulgación para modificar un mensaje de SEI de inicio de segmento de refinamiento progresivo, para incluir la información que indica el valor del POC de la última imagen codificada del segmento de refinamiento progresivo. En algunos ejemplos, el codificador de vídeo 20 puede estar configurado para incluir, en el mensaje de SEI de inicio de segmento de refinamiento progresivo, uno o más elementos sintácticos que indican, directa o indirectamente, el valor del POC de la última imagen codificada en el segmento de refinamiento progresivo.

60 Según una implementación de este tipo, el codificador de vídeo 20 puede generar una sintaxis que indica un valor "delta de POC" para indicar una diferencia entre el valor del POC de la imagen de base y el valor del POC de la última imagen del segmento de refinamiento progresivo. Según otra implementación de ese tipo, el codificador de vídeo 20 puede generar un elemento sintáctico que indica un valor "delta de LSB" que indica una diferencia entre los
65

respectivos bits menos significativos (LSB), de la imagen de base y de la última imagen del segmento de refinamiento progresivo.

5 La sintaxis y la semántica para un ejemplo de un mensaje de SEI de inicio de segmento de refinamiento progresivo, en el que el codificador de vídeo 20 puede incluir un elemento sintáctico delta de POC, según uno o más aspectos de esta divulgación, se describen en la tabla de sintaxis 3 a continuación.

Tabla de sintaxis 3

inicio_de_segmento_de_refinamiento_progresivo (tamañoDeCargaÚtil) {	Descriptor
Identificador_de_refinamiento_progresivo	ue(v)
delta_recuento_orden_imagen	ue(v)
}	

10

En el ejemplo de la tabla de sintaxis 3, el elemento sintáctico delta de POC se denota por "delta_recuento_orden_imagen". En las implementaciones en las que el codificador de vídeo 20 señala el elemento sintáctico delta de POC, el decodificador de vídeo 30 puede utilizar el valor del elemento sintáctico delta de POC señalado para determinar la última imagen codificada del segmento de refinamiento progresivo. Más específicamente, el decodificador de vídeo 30 puede detectar el comienzo del segmento de refinamiento progresivo, en base a la detección del mensaje de SEI de inicio de segmento de refinamiento progresivo. Además, el decodificador de vídeo 30 puede aplicar el valor del elemento sintáctico delta de POC al valor del POC de la imagen de base, para obtener el valor del POC de la última imagen del segmento de refinamiento progresivo. A su vez, el decodificador de vídeo 30 puede aplicar iterativamente la operación de refinamiento progresivo indicada por el valor del elemento sintáctico identificador_de_refinamiento_progresivo a cada una de una serie de imágenes codificadas que siguen a la imagen de base, hasta que el decodificador de vídeo 30 detecte la última imagen del segmento. En particular, el decodificador de vídeo 30 puede detectar la última imagen del segmento de refinamiento progresivo en base al valor del POC de la última imagen, tal como se obtiene en base al valor del elemento sintáctico delta de POC.

25

De manera similar, en las implementaciones donde el codificador de vídeo 20 señala el elemento sintáctico delta de LSB, el decodificador de vídeo 30 puede aplicar el valor del elemento sintáctico delta de LSB a los LSB del valor del POC de la imagen de base. Mediante la aplicación del valor del elemento sintáctico delta de LSB a los LSB del valor del POC de la imagen de base, el decodificador de vídeo 30 puede obtener los LSB del valor del POC de la última imagen codificada en el segmento de refinamiento progresivo. A su vez, utilizando los LSB obtenidos del valor del POC de la última imagen codificada en el segmento de refinamiento progresivo, el decodificador de vídeo 30 puede detectar el límite final del segmento de refinamiento progresivo.

30

Más específicamente, el decodificador de vídeo 30 puede decodificar la serie de imágenes codificadas que constituyen el segmento de refinamiento progresivo, mediante la aplicación iterativa de la operación identificada de refinamiento progresivo, hasta que el decodificador de vídeo 30 detecte la última imagen codificada del segmento. Al detectar la última imagen codificada del segmento de refinamiento progresivo usando los LSB del valor del POC, el decodificador de vídeo 30 puede dejar de decodificar según los parámetros especificados en el mensaje de SEI de inicio de segmento de refinamiento progresivo (a saber, la operación identificada por el valor del elemento sintáctico identificador_de_refinamiento_progresivo).

40

Mediante la implementación de una o más de las técnicas descritas anteriormente, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden implementar la codificación de vídeo utilizando el refinamiento progresivo, soportando al mismo tiempo la ajustabilidad temporal a escala de los flujos de bits de vídeo codificado. Según diversos ejemplos, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden utilizar información estática, tal como los valores del POC, para determinar los límites de un segmento de refinamiento progresivo, independientemente de si el segmento de refinamiento progresivo está o no ajustado temporalmente a escala para incluir un número de imágenes menor que el determinado originalmente.

50

De esta manera, el dispositivo de origen 12 y/o el dispositivo de destino 14 pueden ser un ejemplo de un dispositivo para la codificación de datos de vídeo, que comprende un codificador de vídeo, es decir, el codificador de vídeo 20 y el decodificador de vídeo 30, respectivamente. Además, según las técnicas descritas anteriormente, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden ser ejemplos de un codificador de vídeo configurado para determinar información que indica un valor del recuento de orden de imágenes (POC) de una última imagen entre una pluralidad de imágenes en un segmento de refinamiento progresivo en un flujo de bits codificado, y codificar al menos algunas de las imágenes en el segmento de refinamiento progresivo según una operación de refinamiento progresivo.

55

Además, en algunos ejemplos, la información determinada incluye un valor de delta de POC que indica una diferencia entre un valor del POC de una imagen que precede inmediatamente al segmento de refinamiento progresivo y el valor del POC de la última imagen del segmento de refinamiento progresivo. En algunos ejemplos, la

60

información determinada incluye un delta de los bits menos significativos (LSB), que indica una diferencia entre los LSB de un valor del POC de una imagen que precede inmediatamente al segmento de refinamiento progresivo y los LSB de un valor del POC de la última imagen del segmento de refinamiento progresivo. En algunos ejemplos, la información determinada incluye un valor del POC de la última imagen del segmento de refinamiento progresivo.

Según algunas implementaciones de las técnicas descritas en el presente documento, para determinar la información, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden estar configurados para determinar la información basada, al menos en parte, en un mensaje de información de mejora suplementaria (SEI), incluido en el flujo de bits codificado. En un ejemplo, el mensaje de SEI se compone de un mensaje de SEI de inicio de segmento de refinamiento progresivo que indica un límite inicial del segmento de refinamiento progresivo.

En algunos ejemplos, según las técnicas descritas en el presente documento, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden estar configurados además para determinar que una imagen codificada preceda inmediatamente al segmento de refinamiento progresivo en un flujo de bits de vídeo codificado. En un ejemplo, cada una entre la pluralidad de imágenes en la secuencia de refinamiento progresivo define un refinamiento de la calidad en comparación con la imagen codificada que precede inmediatamente a la secuencia de refinamiento progresivo.

Según algunas implementaciones de las técnicas de esta divulgación, para codificar al menos algunas de las imágenes en el segmento de refinamiento progresivo según la operación de refinamiento progresivo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden estar configurados para aplicar la operación de refinamiento progresivo a cada imagen en el segmento de refinamiento progresivo, a excepción de la última imagen en el segmento de refinamiento progresivo. En algunas implementaciones, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden incluir un circuito integrado y/o un microprocesador y/o un dispositivo de comunicación que comprenda los respectivos codificador de vídeo 20 y/o decodificador de vídeo 30. En algunos ejemplos, el codificador de vídeo 20 puede estar configurado para decodificar datos de vídeo codificados.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 20 que puede implementar las técnicas de codificación de datos de vídeo de acuerdo con uno o más aspectos de esta divulgación. El codificador de vídeo 20 puede realizar la intra-codificación y la inter-codificación de bloques de vídeo dentro de fragmentos de vídeo. La intra-codificación se apoya en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La inter-codificación se apoya en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. La intra-modalidad (modalidad I) puede referirse a cualquiera de varias modalidades de compresión de base espacial. Las inter-modalidades, tales como la predicción unidireccional (modalidad P) o la bi-predicción (modalidad B), pueden referirse a cualquiera de varias modalidades de codificación de base temporal.

Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama de vídeo a codificar. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye la unidad de procesamiento de predicciones 40, la memoria de tramas de referencia 64, el sumador 50, la unidad de procesamiento de transformaciones 52, la unidad de cuantificación 54 y la unidad de codificación por entropía 56. A su vez, la unidad de procesamiento de predicciones 41 incluye la unidad de compensación de movimiento 44, la unidad de estimación de movimiento 42, la unidad de intra-predicción 46 y la unidad de división 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye además una unidad de cuantificación inversa 58, una unidad de transformación inversa 60 y un sumador 62. También puede incluirse un filtro de desbloqueo (no mostrado en la FIG. 2) para filtrar límites de bloque, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de desbloqueo filtrará habitualmente la salida del sumador 62. También pueden usarse filtros adicionales (en bucle o pos-bucle), además del filtro de desbloqueo. Estos filtros no se muestran por razones de brevedad pero, si se desea, pueden filtrar la salida del sumador 62 (tal como un filtro en bucle).

Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o un fragmento de vídeo que va a codificarse. La trama o el fragmento pueden ser divididos en múltiples bloques de vídeo por la unidad de procesamiento de predicciones 40. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 llevan a cabo la codificación inter-predictiva del bloque de vídeo recibido con respecto a uno o más bloques en una o más tramas de referencia, para proporcionar la predicción temporal. La unidad de intra-predicción 46, como alternativa, puede llevar a cabo la codificación intra-predictiva del bloque de vídeo recibido, con respecto a uno o más bloques vecinos en la misma trama o fragmento que el bloque que va a codificarse, para proporcionar la predicción espacial. El codificador de vídeo 20 puede llevar a cabo múltiples pasadas de codificación, por ejemplo, para seleccionar una modalidad de codificación adecuada para cada bloque de datos de vídeo.

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

La unidad de procesamiento de predicciones 40 puede seleccionar una de las modalidades de codificación, intra-codificación o inter-codificación, por ejemplo, en función de los resultados de errores, y proporciona el bloque intra-codificado o inter-codificado resultante al sumador 50, para generar datos de bloque residuales, y al sumador 62, para reconstruir el bloque codificado para su uso como trama de referencia. La unidad de procesamiento de predicciones 40 proporciona además elementos sintácticos, tales como vectores de movimiento, indicadores de intra-modalidad, información de división y otra información sintáctica de este tipo, a la unidad de codificación por entropía 56. La unidad de procesamiento de predicciones 40 puede seleccionar una o más inter-modalidades usando el análisis de distorsión de velocidad.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen actual de vídeo, con respecto a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada), con respecto al bloque actual que está siendo codificado dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque que se revela como estrechamente coincidente con el bloque a codificar, en lo que respecta a la diferencia de píxeles, que puede determinarse mediante la suma de una diferencia absoluta (SAD), una suma de diferencia de cuadrados (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones fraccionarias de píxeles de imágenes de referencia almacenadas en la memoria de tramas de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxeles completos y a las posiciones de fracciones de píxel, y emitir un vector de movimiento con una precisión de fracciones de píxel.

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

La compensación de movimiento, llevada a cabo por la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo en base al vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente, en algunos ejemplos. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que está siendo codificado, generando valores de diferencia de píxel, como se expone posteriormente. En general, la unidad de estimación de movimiento 42 lleva a cabo la estimación de movimiento con respecto a los bloques de codificación de luma, y la unidad de compensación de movimiento 44 usa los vectores de movimiento calculados en base a los bloques de codificación de luma, tanto para los bloques de codificación de croma como para los bloques de codificación de luma. La unidad de procesamiento de predicciones 40 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo, para su uso por parte del decodificador de vídeo 30 en la decodificación de los bloques de vídeo del fragmento de vídeo.

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

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

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

El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modalidad 40 del bloque de vídeo original que está siendo codificado. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformaciones 52 aplica una transformación, tal como una transformación discreta del coseno (DCT) o una transformación conceptualmente similar, al bloque residual, generando un bloque de vídeo que comprende valores residuales de coeficientes de transformación. La unidad de procesamiento de transformaciones 52 puede llevar a cabo otras transformaciones que son conceptualmente similares a la DCT. También podrían usarse transformaciones de ondículas, transformaciones de enteros, transformaciones de sub-bandas u otros tipos de transformaciones. En cualquier caso, la unidad de procesamiento de transformaciones 52 aplica la transformación al bloque residual, generando un bloque de coeficientes de transformación residuales. La transformación puede convertir la información residual, desde un dominio de valores de píxel a un dominio de transformaciones, tal como un dominio de frecuencia. La unidad de procesamiento de transformaciones 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformación para reducir adicionalmente la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos de, o a todos, los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar a continuación un escaneado de la matriz, incluyendo los coeficientes de transformación cuantificados. Como alternativa, la unidad de codificación por entropía 56 puede realizar el escaneado.

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

La unidad de cuantificación inversa 58 y la unidad de transformación inversa 60 aplican la cuantificación inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso posterior como un bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las tramas de la memoria de tramas de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido, para calcular valores de fracciones de píxel para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, generado por la unidad de compensación de movimiento 44 para generar un bloque de vídeo reconstruido para su almacenamiento en la memoria de tramas de referencia 64. El bloque de vídeo reconstruido puede ser usado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para inter-codificar un bloque en una trama de vídeo posterior.

Diversos componentes del codificador de vídeo 20 pueden estar configurados para implementar una o más de las técnicas de esta divulgación, para codificar datos de vídeo utilizando refinamiento progresivo, soportando al mismo tiempo la ajustabilidad temporal a escala de flujos de bits de vídeo. Por ejemplo, el codificador de vídeo 20 puede implementar una o más de las técnicas para generar y señalar un mensaje de información de mejora suplementaria (SEI), de manera que el mensaje de SEI permita a un dispositivo de recepción (por ejemplo, un decodificador de vídeo o componentes del mismo) determinar los límites inicial y final de un segmento de refinamiento progresivo, independientemente de si el segmento está o no temporalmente ajustado a escala para incluir un menor número de imágenes codificadas. En un ejemplo, la unidad de procesamiento de predicciones 40 puede estar configurada para generar un mensaje de SEI de inicio de segmento de refinamiento progresivo, según uno o más aspectos de esta divulgación.

El codificador de vídeo 20 puede estar configurado, según la norma HEVC WD 9, a la norma AVC o a otra norma de codificación de vídeo, para incluir metadatos en un flujo de bits de vídeo codificado. En diversos ejemplos, el codificador de vídeo 20 puede incluir metadatos que no son requeridos por un decodificador para decodificar el flujo

señalizado de bits de vídeo codificado. Como algunos ejemplos, el codificador de vídeo 20 puede señalar metadatos que permiten a un decodificador de vídeo determinar la temporización de salida de imágenes, determinar la información de pantalla asociada a una o más imágenes, detectar la información de pérdidas y ocultar y/o remediar las pérdidas detectadas.

5 Además, el codificador de vídeo 20 puede generar cualquier número de unidades de la capa de abstracción de red (NAL) de SEI en una unidad de acceso (AU) particular señalizada en el flujo de bits de vídeo codificado. A su vez, el codificador de vídeo 20 puede incluir cualquier número de mensajes de SEI en una unidad particular de NAL de SEI. La tabla 1 anterior enumera varios mensajes de SEI que el codificador de vídeo 20 puede generar, y los correspondientes usos / propósitos de los mensajes de SEI enumerados, según el documento HEVC WD9.

10 El codificador de vídeo 20 puede configurarse, o estar operativo de otro modo, para generar y señalar segmentos de refinamiento progresivo en un flujo de bits de vídeo codificado. Como se ha descrito anteriormente con respecto a la FIG. 1, un segmento de refinamiento progresivo, según el documento HEVC WD9, puede incluir una secuencia de imágenes codificadas en orden de decodificación. En algunos ejemplos, la secuencia de imágenes codificadas en el segmento de refinamiento progresivo también puede estar dispuesta según el orden de salida. En particular, cada imagen codificada del segmento de refinamiento progresivo puede representar un refinamiento gradual y acumulativo de la calidad en comparación con una imagen codificada que precede inmediatamente al segmento de refinamiento progresivo en el orden de decodificación.

20 El codificador de vídeo 20 puede señalar un mensaje de SEI de inicio de segmento de refinamiento progresivo para indicar el límite inicial del segmento de refinamiento progresivo. Como se ilustra en la tabla de sintaxis 1 anterior, el codificador de vídeo 20 puede, según el documento HEVC WD9, incluir los elementos sintácticos `identificador_de_refinamiento_progresivo` y `núm_de_etapas_de_refinamientos_menos1` en el mensaje de SEI de inicio de segmento de refinamiento progresivo. Según el documento HEVC WD9, el codificador de vídeo 20 puede determinar que el segmento de refinamiento progresivo continúe (por ejemplo, que cada imagen codificada posterior en el flujo de bits codificado sea únicamente un refinamiento de la calidad con respecto a la imagen codificada anterior), hasta que se cumplan una o más condiciones. Por ejemplo, el codificador de vídeo 20 puede determinar que el segmento de refinamiento progresivo continúe hasta que se cumpla una de las siguientes condiciones:

- 30 – El codificador de vídeo 20 determina el comienzo de una nueva secuencia de vídeo codificada.
- El codificador de vídeo 20 determina el fin del flujo de bits de vídeo codificado.
- 35 – El codificador de vídeo 20 determina que el valor del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` es mayor que cero, y que el bit menos significativo del valor del POC (indicado por `lsb_recuento_orden_imagen`) del siguiente fragmento en el orden de decodificación es igual al resultado de la siguiente fórmula: $(\text{LsbActualRecuentoOrdenImagen} + \text{núm_de_etapas_de_refinamiento_menos1} + 1) \% \text{LsbMáximoRecuentoOrdenImagen}$, donde `LsbActualRecuentoOrdenImagen` es el valor de `lsb_recuento_orden_imagen` de la imagen en la unidad de acceso que contiene el mensaje de SEI. El operador "%" se refiere a una operación de módulo, que produce el valor del resto de una operación de división de enteros.
- 40 – El codificador de vídeo 20 determina que el valor del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` es igual a cero, y que se codifica un mensaje de SEI de fin de segmento de refinamiento progresivo con el mismo valor del `identificador_de_refinamiento_progresivo` que el del mensaje actual de SEI de inicio de segmento de refinamiento progresivo.

50 Además, el codificador de vídeo 20 puede indicar a un decodificador que una unidad de NAL de SEI precede, en el orden de decodificación, a la primera unidad de NAL de la capa de codificación de vídeo (VCL) en una AU actual, mediante la generación de la unidad de NAL de SEI para incluir un mensaje de SEI de inicio de segmento de refinamiento progresivo y un elemento sintáctico `"nuh_reservado_cero_6bits"` con un valor igual a cero. El codificador de vídeo 20 puede establecer el valor del elemento sintáctico `identificador_de_refinamiento_progresivo` dentro de una gama numérica que comienza en 0 y termina en $(2^{32}-2)$, ambos valores inclusive. Por ejemplo, el codificador de vídeo 20 puede fijar el elemento sintáctico `identificador_de_refinamiento_progresivo` en valores en las gamas de 0 a 255, ambos valores inclusive, y de 512 a $(2^{31} - 1)$, ambos valores inclusive, para identificar una operación de refinamiento progresivo específico, asociada al segmento de refinamiento progresivo. En otras palabras, cualquier valor asignado al elemento sintáctico `identificador_de_refinamiento_progresivo` en las gamas descritas anteriormente puede estar asociado a una operación de refinamiento progresivo particular, con soporte en el documento HEVC WD9.

65 Por otra parte, los valores del elemento sintáctico `identificador_de_refinamiento_progresivo` que caen dentro de las gamas de 256 a 511, ambos valores inclusive, y de 2^{31} a $(2^{32} - 2)$, ambos valores inclusive, están reservados para uso futuro por la ITU- T y/o la ISO / IEC. Más específicamente, si el codificador de vídeo 20 fija el elemento sintáctico `identificador_de_refinamiento_progresivo` en un valor en una de las gamas reservadas, descritas

anteriormente, un dispositivo que recibe el flujo de bits, tal como un decodificador de vídeo, puede no tener en cuenta el elemento sintáctico `identificador_de_refinamiento_progresivo`. En particular, para no tener en cuenta este elemento sintáctico, un decodificador de vídeo puede quitar el elemento sintáctico `identificador_de_refinamiento_progresivo` del flujo recibido de bits de vídeo codificado, y desechar el elemento sintáctico `identificador_de_refinamiento_progresivo`.

Como se ha descrito, el codificador de vídeo 20 y/o los componentes del mismo pueden configurarse para soportar la ajustabilidad temporal a escala de flujos de bits de vídeo codificado, tal como según el documento HEVC WD9. Por ejemplo, el codificador de vídeo 20 puede generar un flujo completo de bits de vídeo codificado, del que un dispositivo de recepción, tal como un dispositivo de decodificación o un dispositivo intermedio, puede extraer un sub-flujo de bits. Por ejemplo, un dispositivo intermedio, como por ejemplo un servidor de flujos de transmisión, o un elemento de red consciente de los medios ("MANE"), puede extraer un subconjunto temporal de imágenes codificadas del conjunto completo de imágenes codificadas, incluidas en el flujo completo de bits de vídeo codificado. En algunos ejemplos, el subconjunto temporal puede representar un verdadero subconjunto del conjunto completo de imágenes codificadas incluidas en el flujo completo de bits de vídeo codificado. Según estos ejemplos, el flujo completo de bits de vídeo codificado puede incluir cada imagen codificada del subconjunto temporal, y al menos una imagen codificada adicional que no esté incluida en el subconjunto temporal.

Para soportar diversas velocidades de imagen según la ajustabilidad temporal a escala, un dispositivo intermedio puede estar configurado para extraer subconjuntos temporales de diferentes recuentos de imágenes, desde el conjunto completo de imágenes codificadas incluidas en el flujo completo de bits de vídeo codificado. Cada subconjunto temporal diferente, extraído por el dispositivo intermedio (por ejemplo, para soportar diferentes velocidades de imagen), puede representar un subconjunto temporal, o sub-flujo de bits, independientemente decodificable. En otras palabras, un decodificador de vídeo que recibe un sub-flujo de bits temporalmente ajustado a escala, extraído del flujo completo de bits de vídeo codificado, puede decodificar el subconjunto temporal de imágenes codificadas sin ningún dato adicional, tal como la información incluida en el flujo completo de bits de vídeo codificado, pero excluida del sub-flujo de bits.

Un flujo completo de bits de vídeo codificado, generado por el codificador de vídeo 20, según el documento HEVC WD9, puede incluir varias sub-capas temporales. Además, cada unidad de NAL generada por el codificador de vídeo 20 puede pertenecer a una sub-capa específica, según lo indicado por un valor correspondiente de "IdentificadorTemporal". Por ejemplo, el codificador de vídeo 20 puede establecer el valor del `IdentificadorTemporal` de una unidad de NAL igual al valor del correspondiente elemento sintáctico "`identificador_temporal_más1`", menos uno. Además, el codificador de vídeo 20 puede determinar que todas las unidades de NAL de `VCL` de una sola imagen pertenecen a una sola sub-capa (es decir, la misma sub-capa). En otras palabras, el codificador de vídeo 20 puede codificar una imagen, de tal manera que la propia imagen codificada pertenezca a la sub-capa específica correspondiente a las unidades de NAL asociadas a la imagen codificada.

Por ejemplo, según el documento HEVC WD9, el codificador de vídeo 20 puede generar un flujo de bits de vídeo codificado de tal manera que un proceso de decodificación de una sub-capa inferior del flujo de bits no dependa de los datos en una sub-capa superior del flujo de bits. Además, el dispositivo intermedio puede generar un sub-flujo de bits a partir de un flujo de bits completo, que sea conforme al documento HEVC WD9, mediante la eliminación, del flujo completo de bits, de todas las unidades de NAL asociadas a un valor de `IdentificadorTemporal` que sea superior a un valor específico. A su vez, el sub-flujo de bits generado de esta manera puede representar por sí mismo un flujo de bits conforme al documento HEVC WD9. El codificador de vídeo 20 y/o uno o más componentes del mismo pueden garantizar que se cumplan todas las condiciones para la conformidad del flujo de bits (por ejemplo, restricciones de almacenamiento temporal), con respecto al documento HEVC WD9, para el flujo completo de bits de vídeo codificado, y para cualquier sub-capa determinada del mismo.

Como se ha descrito, al ajustar temporalmente a escala un flujo completo de bits de vídeo codificado, el dispositivo intermedio puede extraer un subconjunto temporal de imágenes codificadas a partir del flujo completo de bits de vídeo codificado. Por ejemplo, el subconjunto temporal puede ser un verdadero subconjunto de las imágenes codificadas señalizadas en el flujo completo de bits de vídeo codificado y, por lo tanto, el dispositivo intermedio puede eliminar una o más imágenes codificadas del flujo completo de bits codificado para generar el sub-flujo de bits. En los ejemplos, el dispositivo intermedio puede eliminar una o más imágenes codificadas que se incluyen en un segmento de refinamiento progresivo. Sin embargo, en estos ejemplos, el dispositivo intermedio puede no estar configurado para actualizar los datos señalizados en el mensaje de SEI de inicio de segmento de refinamiento progresivo, para reflejar el cambio (es decir, la reducción) en el número de imágenes codificadas incluidas en el segmento de refinamiento progresivo temporalmente ajustado a escala. En otras palabras, el dispositivo intermedio puede, potencialmente, señalar un valor incorrecto del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` a un dispositivo receptor, tal como un decodificador de vídeo. A su vez, mediante la comunicación de un valor incorrecto del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1`, el dispositivo intermedio puede hacer que el decodificador de vídeo aplique la operación de refinamiento progresivo para decodificar imágenes después del final del segmento de refinamiento progresivo temporalmente ajustado a escala.

Para mitigar o eliminar la decodificación errónea de las imágenes que se sitúan después del final del segmento de refinamiento progresivo en el flujo de bits de vídeo codificado, el codificador de vídeo 20 puede implementar una o más técnicas de esta divulgación. Según algunas implementaciones de las técnicas, el codificador de vídeo 20 puede generar el mensaje de SEI de inicio de segmento de refinamiento progresivo para incluir información que permita a un decodificador de vídeo determinar el valor del recuento de orden de imágenes (POC) de la última imagen del segmento de refinamiento progresivo. El valor del POC puede representar un valor estático que se asocia a una imagen codificada en particular, señalizada en el flujo de bits de vídeo codificado y, por lo tanto, puede identificar la imagen codificada en particular, independientemente del ajuste temporal a escala.

Por ejemplo, al obtener el valor del POC de la última imagen codificada del segmento de refinamiento progresivo, un decodificador de vídeo, que recibe el flujo de bits de vídeo codificado, puede identificar con precisión la última imagen codificada. Más específicamente, el decodificador de vídeo puede identificar la última imagen codificada del segmento de refinamiento progresivo, independientemente de si el dispositivo intermedio ha ajustado temporalmente a escala, o no, el segmento de refinamiento progresivo. En ejemplos donde el dispositivo intermedio ajusta temporalmente a escala el segmento de refinamiento progresivo, el decodificador de vídeo puede identificar con precisión la última imagen del segmento de refinamiento progresivo, sin que importe el grado en que el dispositivo intermedio ajusta temporalmente a escala el segmento de refinamiento progresivo. En otras palabras, el decodificador de vídeo puede identificar la última imagen codificada (es decir, el límite final) del segmento de refinamiento progresivo, independientemente de la velocidad de imagen que el dispositivo intermedio genera, utilizando la ajustabilidad temporal a escala.

Según diversas implementaciones de las técnicas descritas en el presente documento, el codificador de vídeo 20 puede generar un mensaje de SEI de inicio de segmento de refinamiento progresivo para incluir información que un decodificador puede utilizar para determinar el valor del POC asociado a la última imagen codificada del segmento de refinamiento progresivo. En una implementación, el codificador de vídeo 20 puede generar el mensaje de SEI de inicio de segmento de refinamiento progresivo para incluir datos que indican un valor de "delta de POC". Más específicamente, el codificador de vídeo 20 puede generar el valor de delta de POC, de manera que el valor de delta de POC indique una diferencia entre el valor del POC de la imagen de base y el valor del POC de la última imagen codificada del segmento de refinamiento progresivo.

Un decodificador de vídeo que recibe un flujo de bits de vídeo codificado, generado por el codificador de vídeo 20, puede utilizar el mensaje de SEI de inicio de segmento de refinamiento progresivo para detectar el límite inicial (por ejemplo, la primera imagen codificada) de un segmento correspondiente de refinamiento progresivo. Además, el decodificador de vídeo puede determinar el límite final (por ejemplo, la última imagen codificada) del segmento de refinamiento progresivo, aplicando el valor de delta de POC al valor del POC de la imagen de base (por ejemplo, la imagen codificada que precede inmediatamente al límite inicial, en el orden de decodificación). Por ejemplo, mediante la aplicación del valor de delta de POC al valor del POC de la imagen de base, el decodificador de vídeo puede obtener el valor del POC del límite final del segmento de refinamiento progresivo.

Como se ha descrito, el valor del POC de una imagen codificada en particular puede ser un valor estático asociado a la imagen codificada, y puede indicar la posición de la imagen codificada en el orden de decodificación. Al proporcionar un decodificador con datos de los cuales obtener el valor del POC del límite final del segmento de refinamiento progresivo, el codificador de vídeo 20 puede permitir al decodificador determinar el límite final del segmento de refinamiento progresivo, con independencia de si, y en qué magnitud, un dispositivo intermedio ajusta temporalmente a escala, o no, el segmento de refinamiento progresivo. El codificador de vídeo 20 puede indicar el valor de delta de POC en forma de un elemento sintáctico "delta_recuento_orden_imagen", como se ilustra en la tabla de sintaxis 3 anterior. En diversos ejemplos, el codificador de vídeo 20 puede señalar el elemento sintáctico delta_recuento_orden_imagen en el mensaje de SEI de inicio de segmento de refinamiento progresivo, o en una o más cabeceras de fragmento, asociadas a los respectivos fragmentos de la imagen de base. De esta manera, el codificador de vídeo 20 puede implementar una o más técnicas de esta divulgación para permitir a un decodificador decodificar una secuencia de refinamiento progresivo utilizando una correspondiente operación de refinamiento progresivo, soportando a la vez la ajustabilidad temporal a escala de flujos de bits de vídeo codificado.

Según algunas implementaciones en las que el codificador de vídeo 20 genera el mensaje de SEI de inicio de segmento de refinamiento progresivo para incluir datos asociados al valor del POC del límite final del segmento de refinamiento progresivo, el codificador de vídeo 20 puede incluir un delta de los bits menos significativos (LSB) en el mensaje de SEI de inicio de segmento de refinamiento progresivo. En estas implementaciones, el codificador de vídeo 20 puede generar el delta de los LSB para indicar una diferencia entre el valor de los LSB del POC de la imagen de base, y el valor de los LSB del valor del POC de la última imagen codificada del segmento de refinamiento progresivo.

A su vez, el codificador de vídeo 20 puede permitir a un decodificador que recibe el flujo de bits de vídeo codificado obtener el valor del POC del límite final del segmento de refinamiento progresivo mediante la aplicación del delta de los LSB a los LSB del valor del POC de la imagen de base. Por ejemplo, el codificador de vídeo 20 puede generar el mensaje de SEI de inicio de segmento de refinamiento progresivo para incluir un elemento sintáctico que indica el delta de los LSB, en lugar del elemento sintáctico "núm_de_etapas_de_refinamiento_menos1". Según algunas

implementaciones, el codificador de vídeo 20 puede señalar el elemento sintáctico como parte de una o más cabeceras de fragmento, asociadas a los respectivos fragmentos de la imagen de base.

Al incluir el elemento sintáctico para indicar el delta de los LSB, la unidad de codificación por entropía 56 puede permitir al decodificador determinar el límite final del segmento de refinamiento progresivo, con independencia de si, y en qué magnitud, el dispositivo intermedio ajusta temporalmente a escala, o no, el segmento de refinamiento progresivo. Por ejemplo, señalizando el delta de los LSB, el codificador de vídeo 20 puede permitir a un decodificador obtener el valor del POC del límite final del segmento de refinamiento progresivo y, por lo tanto, identificar el límite final usando un valor estático asociado al límite final. De esta manera, el codificador de vídeo 20 puede implementar una o más técnicas de esta divulgación para permitir a un decodificador decodificar una secuencia de refinamiento progresivo utilizando una correspondiente operación de refinamiento progresivo, soportando al mismo tiempo la ajustabilidad temporal a escala de los flujos de bits de vídeo codificado.

Según otras implementaciones de las técnicas descritas en el presente documento, el codificador de vídeo 20 puede generar el mensaje de SEI de inicio de segmento de refinamiento progresivo para incluir datos que indican el número de imágenes codificadas consecutivas que 1) están asociadas a un valor de IdentificadorTemporal que es menor o igual al valor de IdentificadorTemporal de la imagen de base y 2) representan refinamientos progresivos de calidad de la imagen de base. En algunos ejemplos, un valor decreciente de IdentificadorTemporal puede indicar un avance en el flujo de bits de vídeo codificado, según el orden de decodificación. Por ejemplo, la primera imagen codificada del segmento de refinamiento progresivo puede tener un valor de IdentificadorTemporal que es menor que el valor de IdentificadorTemporal de la imagen de base. A su vez, la segunda imagen codificada del segmento de refinamiento progresivo puede tener un valor de IdentificadorTemporal que es menor que el IdentificadorTemporal de la primera imagen codificada del segmento de refinamiento progresivo, y así sucesivamente.

La imagen de base puede ser la imagen codificada incluida en la unidad de acceso actual y la imagen de base puede preceder inmediatamente a la primera imagen codificada del segmento de refinamiento progresivo, en el orden de decodificación. En algunos ejemplos, la imagen de base también puede preceder inmediatamente a la primera imagen codificada del segmento de refinamiento progresivo, en el orden de salida. En algunos de tales ejemplos, el segmento de refinamiento progresivo puede incluir una secuencia de imágenes codificadas que son consecutivas, tanto en el orden de decodificación como en el orden de salida.

Según estas implementaciones de las técnicas descritas en el presente documento, el codificador de vídeo 20 puede generar el mensaje de SEI de inicio de segmento de refinamiento progresivo para incluir datos que indican el número de imágenes codificadas consecutivas que tienen un valor de IdentificadorTemporal menor que el valor de IdentificadorTemporal de la imagen de base, así como representar refinamientos progresivos de la calidad de la imagen de base. Por ejemplo, la información puede indicar el número de imágenes codificadas en la secuencia que conforma el segmento de refinamiento progresivo. En estas implementaciones, un dispositivo intermedio puede determinar el número de imágenes codificadas consecutivas que cumplen las condiciones enumeradas anteriormente. Por ejemplo, el dispositivo intermedio puede determinar un número de imágenes codificadas consecutivas que cumplen las condiciones, con respecto a cada subconjunto temporal de imágenes codificadas, extraídas del flujo completo de bits de vídeo codificado.

Al determinar el número de imágenes codificadas consecutivas que satisfacen las condiciones anteriores, el dispositivo intermedio puede determinar con mayor precisión el número de imágenes codificadas disponibles en el segmento de refinamiento progresivo, incluso en escenarios donde el dispositivo intermedio ajusta temporalmente a escala el segmento de refinamiento progresivo. A su vez, el dispositivo intermedio puede comunicar el número determinado de imágenes codificadas consecutivas que tienen un valor de IdentificadorTemporal menor que el valor de IdentificadorTemporal de la imagen de base, así como representar refinamientos progresivos de calidad de la imagen de base. En diversos ejemplos, el codificador de vídeo 20 puede generar el mensaje de SEI de inicio de segmento de refinamiento progresivo para incluir un elemento sintáctico que indica el número determinado de imágenes codificadas consecutivas que tienen un valor de IdentificadorTemporal menor que el valor de IdentificadorTemporal de la imagen de base, así como representar refinamientos progresivos de calidad de la imagen de base. Por ejemplo, el codificador de vídeo 20 puede generar el mensaje de SEI de inicio de segmento de refinamiento progresivo para incluir el elemento sintáctico que indica el número determinado, en lugar del elemento sintáctico `núm_de_etapas_derefinamiento_menos1`, ilustrado en la tabla de sintaxis 1 anterior. En algunos ejemplos, el codificador de vídeo 20 puede señalar el elemento sintáctico que indica el número determinado como parte de una o más cabeceras de fragmento, asociadas a los respectivos fragmentos de la imagen de base.

El codificador de vídeo 20 puede, señalizando el número de imágenes codificadas consecutivas que tienen un valor de IdentificadorTemporal menor que el valor de IdentificadorTemporal de la imagen de base y que representan refinamientos progresivos de calidad de la imagen de base, proporcionar un decodificador de vídeo con una longitud exacta del segmento de refinamiento progresivo. Como ejemplo, señalizando la información descrita anteriormente como parte de un mensaje de SEI de inicio de segmento de refinamiento progresivo, el codificador de vídeo 20 puede proporcionar a un decodificador receptor la longitud del segmento de refinamiento progresivo, independientemente de si un dispositivo intermedio ajusta temporalmente a escala, o no, el segmento de

refinamiento progresivo. A su vez, el decodificador que recibe el mensaje de SEI de inicio de segmento de refinamiento progresivo puede utilizar el elemento sintáctico incluido en el mensaje de SEI de inicio de segmento de refinamiento progresivo para determinar la longitud del segmento de refinamiento progresivo, independientemente de si, o en qué magnitud, el decodificador ajusta temporalmente a escala, o no, el segmento de refinamiento progresivo. De esta manera, el codificador de vídeo 20 puede implementar una o más técnicas de esta divulgación para señalar una serie de imágenes codificadas consecutivas que 1) tienen un valor de IdentificadorTemporal menor que el IdentificadorTemporal de la imagen de base y 2) representan refinamientos progresivos de calidad de la imagen de base, permitiendo de ese modo que un dispositivo de decodificación de vídeo decodifique un segmento de refinamiento progresivo, soportando al mismo tiempo flujos de bits de vídeo ajustables temporalmente a escala.

Como se ha descrito anteriormente con respecto a la FIG. 2, el codificador de vídeo 20 y/o los componentes del mismo pueden realizar un procedimiento de codificación de datos de vídeo, incluyendo el procedimiento la codificación de al menos algunas imágenes entre una pluralidad de imágenes en un segmento de refinamiento progresivo, según una operación de refinamiento progresivo, y la generación de información que indique un valor del recuento de orden de imágenes (POC) de una última imagen del segmento de refinamiento progresivo en un flujo de bits codificado. En algunos ejemplos de implementaciones del procedimiento descrito anteriormente con respecto al codificador de vídeo 20, la información generada incluye un valor de delta de POC que indica una diferencia entre un valor del POC de una imagen que precede inmediatamente al segmento de refinamiento progresivo y el valor del POC de la última imagen del segmento de refinamiento progresivo. De acuerdo con algunos ejemplos de implementaciones del procedimiento descrito anteriormente con respecto al codificador de vídeo 20, la información generada incluye un delta de los bits menos significativos (LSB), que indica una diferencia entre los LSB de un valor del POC de una imagen que precede inmediatamente al segmento de refinamiento progresivo y los LSB de un valor del POC de la última imagen del segmento de refinamiento progresivo. En un ejemplo del procedimiento descrito anteriormente con respecto al codificador de vídeo 20, la información generada incluye un valor del POC de la última imagen del segmento de refinamiento progresivo.

Según algunas implementaciones del procedimiento descrito anteriormente con respecto al codificador de vídeo 20, la generación de la información comprende incluir la información en un mensaje de información de mejora suplementaria (SEI), a señalar en el flujo de bits de vídeo codificado. En una implementación de ese tipo, el mensaje de SEI comprende un mensaje de SEI de inicio de segmento de refinamiento progresivo que indica un límite inicial del segmento de refinamiento progresivo. Según algunas implementaciones del procedimiento descrito anteriormente con respecto al codificador de vídeo 20, el procedimiento puede incluir además la determinación de que una imagen codificada precede inmediatamente al segmento de refinamiento progresivo en un flujo de bits de vídeo codificado. En una implementación de ese tipo, cada imagen en la secuencia de refinamiento progresivo define un refinamiento de la calidad en comparación con la imagen codificada que precede inmediatamente a la secuencia de refinamiento progresivo.

En algunos ejemplos del procedimiento descrito anteriormente con respecto al codificador de vídeo 20, el procedimiento puede comprender además detectar la última imagen entre la pluralidad de imágenes en el segmento de refinamiento progresivo, y dejar de codificar según la operación de refinamiento progresivo. Según algunas implementaciones del procedimiento descrito anteriormente con respecto al codificador de vídeo 20, la codificación de al menos algunas de las imágenes en el segmento de refinamiento progresivo, según la operación de refinamiento progresivo, comprende aplicar la operación de refinamiento progresivo a cada imagen en el segmento de refinamiento progresivo, excepto la última imagen en el segmento de refinamiento progresivo.

En algunos ejemplos, el codificador de vídeo 20 puede estar incluido en un dispositivo para codificar datos de vídeo, tal como un ordenador de sobremesa, un ordenador plegable (es decir, portátil), un ordenador de tableta, un decodificador, un equipo telefónico portátil tal como un denominado teléfono "inteligente", un denominado panel "inteligente", un televisor, una cámara, un dispositivo de visualización, un reproductor de medios digitales, una consola de videojuegos, un dispositivo de flujos de transmisión de vídeo o similares. En estos, u otros, ejemplos, un dispositivo de este tipo para la codificación de datos de vídeo puede incluir uno o más entre un circuito integrado, un microprocesador y un dispositivo de comunicación que incluye el codificador de vídeo 20. En algunos ejemplos, el codificador de vídeo 20 también puede estar configurado para decodificar datos de vídeo codificados, como mediante la decodificación por entropía de los datos de vídeo codificado.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 30 que puede implementar las técnicas de decodificación de datos de vídeo, según uno o más aspectos de esta divulgación. En el ejemplo de la FIG. 3, el decodificador de vídeo 30 incluye una unidad de decodificación por entropía 70, una unidad de compensación de movimiento 72, una unidad de intra-predicción 74, una unidad de procesamiento de cuantificación inversa 76, una unidad de procesamiento de transformación inversa 78, un sumador 80 y una memoria de imágenes de referencia 82. En el ejemplo de la FIG. 2, el decodificador de vídeo 30 incluye la unidad de predicción 71, la cual, a su vez, incluye la unidad de compensación de movimiento 72 y la unidad de intra-predicción 74. En algunos ejemplos, el decodificador de vídeo 30 puede llevar a cabo una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción basados en vectores de movimiento recibidos desde la unidad de decodificación por entropía 70, mientras que la unidad de intra-predicción 74 puede generar datos de predicción

basados en indicadores de modalidad de intra-predicción recibidos desde la unidad de decodificación por entropía 70.

5 En la implementación ilustrada en la FIG. 3, el decodificador de vídeo 30 está acoplado a un elemento de red 68. En diversos ejemplos, el elemento de red 68 puede incluir, ser, o formar parte de, una amplia variedad de dispositivos, tales como un elemento de red consciente de los medios (o "MANE"), un servidor de flujos de transmisión o un dispositivo de extremo de cabecera de red. Por ejemplo, el elemento de red 68 puede estar configurado para recibir un flujo de bits de vídeo codificado, señalizado por el codificador de vídeo 20, y ajustar temporalmente a escala el flujo de bits de vídeo codificado. En este ejemplo, el elemento de red 68 puede retransmitir el flujo de bits, ajustado 10 temporalmente a escala, al decodificador de vídeo 30.

15 Como ejemplo, el elemento de red 68 puede extraer un subconjunto temporal de imágenes codificadas provenientes del conjunto completo de imágenes codificadas, incluidas en el flujo recibido de bits de vídeo codificado. El flujo de bits de vídeo codificado, recibido por el elemento de red 68, puede denominarse en este documento un "flujo completo de bits de vídeo codificado". Además, el subconjunto temporal extraído por el elemento de red 68 puede representar un verdadero subconjunto del conjunto completo de imágenes codificadas incluidas en el flujo completo de bits de vídeo codificado. En otras palabras, el flujo completo de bits de vídeo codificado, recibido por el elemento de red 68, puede incluir cada imagen codificada del subconjunto temporal, y al menos una imagen codificada 20 adicional que no esté incluida en el subconjunto temporal.

25 Para soportar diversas velocidades de imagen, según la ajustabilidad temporal a escala, el elemento de red 68 puede estar configurado para extraer subconjuntos temporales de diferentes recuentos de imágenes, a partir del conjunto completo de imágenes codificadas incluidas en el flujo completo de bits de vídeo codificado. Cada subconjunto temporal diferente, extraído por el elemento de red 68 (por ejemplo, para soportar diferentes velocidades de imagen), puede representar un subconjunto, o sub-flujo de bits, temporal, independientemente decodificable. En otras palabras, un dispositivo, tal como el decodificador de vídeo 30, que recibe un sub-flujo de bits ajustado temporalmente a escala, extraído por el elemento de red 68, puede decodificar el subconjunto temporal de imágenes codificadas sin ningún dato adicional, tal como la información incluida en el flujo completo de bits de vídeo 30 codificado, pero excluida del sub-flujo de bits.

35 El elemento de red 68 puede determinar que un flujo completo de bits de vídeo codificado, señalizado por el codificador de vídeo 20, según el documento HEVC WD9, incluye varias sub-capas temporales. Además, el elemento de red 68 puede determinar que cada unidad de NAL señalizada por el codificador de vídeo 20 pertenece a una sub-capa específica, según lo indicado por un valor correspondiente de "IdentificadorTemporal". Por ejemplo, el elemento de red 68 puede determinar que el valor de IdentificadorTemporal de una unidad de NAL es igual al valor del correspondiente elemento sintáctico "identificador_temporal_más1", menos uno. Además, en este ejemplo, el elemento de red 68 puede determinar que todas las unidades de NAL de VCL de una sola imagen pertenecen a una sola sub-capa (es decir, la misma sub-capa). En otras palabras, el elemento de red 68 puede determinar que una propia imagen específica codificada pertenece a la sub-capa específica correspondiente a las unidades de NAL 40 asociadas a la imagen codificada.

45 Por ejemplo, según el documento HEVC WD9, el codificador de vídeo 20 puede generar un flujo de bits de vídeo codificado de tal manera que un proceso de decodificación de una sub-capa inferior del flujo de bits (por ejemplo, según lo extraído por el elemento de red 68) no dependa de los datos en una sub-capa superior del flujo de bits. El elemento de red 68 puede extraer un sub-flujo de bits a partir de un flujo de bits completo, que es conforme al documento HEVC WD9, mediante la eliminación, del flujo de bits completo, de todas las unidades de NAL asociadas a un valor de IdentificadorTemporal que es mayor que un valor específico. A su vez, el sub-flujo de bits, extraído por el elemento de red 68 de esta manera, puede representar por sí mismo un flujo de bits conforme al documento HEVC WD9. El codificador de vídeo 20 y/o uno o más componentes del mismo pueden garantizar que todas las 50 condiciones para la conformidad del flujo de bits (por ejemplo, restricciones de almacenamiento temporal) con respecto al documento HEVC WD9 se cumplan para cada sub-flujo de bits.

55 Como se ha descrito, en el ajuste temporal a escala de un flujo completo de bits de vídeo codificado, el elemento de red 68 puede extraer un subconjunto temporal de imágenes codificadas a partir del flujo completo de bits de vídeo codificado. Por ejemplo, el subconjunto temporal puede ser un verdadero subconjunto de las imágenes codificadas señalizadas en el flujo completo de bits de vídeo codificado y, por lo tanto, el elemento de red 68 puede eliminar una o más imágenes codificadas del flujo completo de bits codificado para generar el sub-flujo de bits. En los ejemplos, el elemento de red 68 puede eliminar una o más imágenes codificadas que se incluyen en un segmento de refinamiento progresivo. 60

65 Sin embargo, en estos ejemplos, el elemento de red 68 puede no estar configurado para actualizar los datos señalizados en el mensaje de SEI de inicio de segmento de refinamiento progresivo, para reflejar el cambio (es decir, la reducción) en el número de imágenes codificadas incluidas en el segmento de refinamiento progresivo ajustado temporalmente a escala. En otras palabras, el elemento de red 68 puede, potencialmente, retransmitir un valor incorrecto del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` al decodificador de vídeo 30. A su vez, al retransmitir un valor incorrecto del elemento sintáctico `núm_de_etapa_de_refinamiento_menos1`, el elemento

de red 68 puede hacer que el decodificador de vídeo 30 aplique la operación de refinamiento progresivo para decodificar imágenes después del final del segmento de refinamiento progresivo ajustado temporalmente a escala.

5 Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de decodificación por entropía 70 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modalidad de intra-predicción, y otros elementos sintácticos. La unidad de decodificación por entropía 70 remite los vectores de movimiento y otros elementos sintácticos a la unidad de compensación de movimiento 72. El decodificador de vídeo 10 30 puede recibir los elementos sintácticos al nivel del fragmento de vídeo y/o al nivel del bloque de vídeo.

15 Cuando el fragmento de vídeo se codifica como un fragmento intra-codificado (I), la unidad de intra-predicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, en base a una modalidad de intra-predicción señalizada, y datos de bloques previamente decodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento inter-codificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, en base a los vectores de movimiento y a otros elementos sintácticos recibidos desde la unidad de decodificación por entropía 70. Los bloques predictivos pueden ser generados a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de tramas de referencia, la 20 Lista 0 y la Lista 1, usando técnicas de construcción por omisión, en base a las imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

25 La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos para el bloque de vídeo actual que está siendo decodificado. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, intra-predicción o inter-predicción), usada para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de inter-predicción (por ejemplo, fragmento B, fragmento P o fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia del fragmento, vectores de movimiento para cada bloque de vídeo inter-codificado del fragmento, el estado de inter- 30 predicción para cada bloque de vídeo inter-codificado del fragmento y otra información, para decodificar los bloques de vídeo en el fragmento de vídeo actual.

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

40 La unidad de cuantificación inversa 76 cuantifica de manera inversa, es decir, descuantifica, los coeficientes de transformación cuantificados, proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 70. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QPY, calculado por el codificador de vídeo 30 para cada bloque de vídeo en el fragmento de vídeo, para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse.

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

Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual, en base a los vectores de movimiento y a otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales procedentes de la unidad de transformación inversa 78 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72. El 55 sumador 80 representa el componente o los componentes que llevan a cabo esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques decodificados, con el fin de eliminar distorsiones de efecto pixelado. Otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) también pueden ser usados para allanar las transiciones de píxeles, o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo decodificados en una trama o imagen dada son a continuación almacenados en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para la posterior 60 compensación de movimiento. La memoria de imágenes de referencia 82 almacena también vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

65 El decodificador de vídeo 30, y varios componentes del mismo, pueden implementar las técnicas de esta divulgación para decodificar un segmento de refinamiento progresivo, soportando al mismo tiempo los flujos de bits de vídeo ajustables temporalmente a escala. Como ejemplo, la unidad de decodificación por entropía 70 puede implementar

una o más funcionalidades descritas a continuación con respecto al decodificador de vídeo 30. Como se ha descrito, el decodificador de vídeo 30 puede recibir un flujo de bits de vídeo codificado, señalado por un codificador de vídeo. En diversos ejemplos, el decodificador de vídeo 30 puede recibir un flujo completo de bits de vídeo codificado, o un sub-flujo de bits, que el elemento de red 68 puede extraer, según la ajustabilidad temporal a escala. Más específicamente, un sub-flujo de bits ajustado temporalmente a escala puede incluir un subconjunto de imágenes codificadas extraídas del conjunto de imágenes codificadas incluidas en el flujo completo de bits de vídeo codificado. Un subconjunto de imágenes, extraído por el elemento de red 68, según la ajustabilidad temporal a escala, puede ser mencionado en este documento como un "subconjunto temporal." En algunos ejemplos, el subconjunto temporal extraído por el elemento de red 68 puede representar un verdadero subconjunto de las imágenes codificadas en el flujo completo de bits de vídeo codificado. En otras palabras, según estos ejemplos, el flujo completo de bits de vídeo codificado puede incluir cada imagen codificada del subconjunto temporal, y al menos una imagen codificada adicional que no esté incluida en el subconjunto temporal.

Además, según la norma HEVC WD 9, la norma AVC u otras normas de codificación de vídeo, el decodificador de vídeo 30 puede estar configurado, o ser operativo de otra manera, para decodificar los metadatos incluidos en el flujo recibido de bits de vídeo codificado. En varios ejemplos, según el documento HEVC WD9, el decodificador de vídeo 30 puede decodificar metadatos que no son necesarios para la decodificación de las imágenes codificadas señalizadas en el flujo de bits codificado. En diversos ejemplos, el decodificador de vídeo 30 puede decodificar los metadatos para determinar una o más entre las temporizaciones de salida de imágenes, y mostrar información asociada a una o más imágenes. En estos y otros ejemplos, el decodificador de vídeo 30 puede decodificar metadatos para detectar información de pérdidas, y para ocultar y/o remediar una o más pérdidas detectadas.

En algunos ejemplos, por ejemplo, según el documento HEVC WD9, el decodificador de vídeo 30 puede decodificar una o más unidades de la capa de abstracción de red (NAL) de la información de mejora complementaria (SEI) en una determinada unidad de acceso (AU) señalizada en el flujo recibido de bits de vídeo codificado. Además, el decodificador de vídeo 30 puede decodificar uno o más mensajes de SEI incluidos en una única unidad de NAL de SEI que se señala en el flujo recibido de bits de vídeo codificado. La tabla 1 anterior enumera ejemplos de diversos mensajes de SEI que el decodificador de vídeo 30 puede recibir y decodificar (por ejemplo, utilizando la unidad de decodificación por entropía 70), y los correspondientes usos / propósitos de los mensajes de SEI enumerados, según el documento HEVC WD9.

Además, el decodificador de vídeo 30 puede estar configurado, o ser operativo de otra manera, para decodificar segmentos de refinamiento progresivo señalizados en un flujo recibido de bits de vídeo codificado. Como se ha descrito anteriormente con respecto a la FIG. 1, un segmento de refinamiento progresivo, según el documento HEVC WD9, puede incluir una secuencia de imágenes codificadas en orden de decodificación. En algunos ejemplos, la secuencia de imágenes codificadas en el segmento de refinamiento progresivo también puede estar dispuesta según el orden de salida. En diversos escenarios, cada imagen codificada del segmento de refinamiento progresivo puede representar un refinamiento de calidad con respecto a una imagen de base (por ejemplo, una imagen codificada que precede inmediatamente al límite inicial, en el orden de decodificación). En uno de tales escenarios, cada imagen codificada del segmento de refinamiento progresivo puede representar un refinamiento gradual y acumulativo de la calidad en comparación con una imagen codificada que precede inmediatamente al segmento de refinamiento progresivo, en el orden de decodificación.

El decodificador de vídeo 30 puede decodificar un mensaje de SEI de inicio de segmento de refinamiento progresivo, como mediante la implementación de una o más funcionalidades proporcionadas por la unidad de decodificación por entropía 70. En base al mensaje decodificado de SEI de inicio de segmento de refinamiento progresivo, el decodificador de vídeo 30 puede detectar el límite inicial del segmento de refinamiento progresivo. Como se ilustra en la tabla de sintaxis 1 anterior, el decodificador de vídeo 30 puede, según el documento HEVC WD9, decodificar los elementos sintácticos `identificador_de_refinamiento_progresivo` y `núm_de_etapas_de_refinamiento_menos1` en el mensaje señalizado de SEI de inicio de segmento de refinamiento progresivo. Según el documento HEVC WD9, el decodificador de vídeo 30 puede determinar que el segmento de refinamiento progresivo continúa (por ejemplo, que cada imagen codificada posterior en el flujo de bits codificado es únicamente un refinamiento de la calidad con respecto a la imagen codificada anterior), hasta que se cumplan una o más condiciones. Por ejemplo, el decodificador de vídeo 30 puede determinar que el segmento de refinamiento progresivo continúa hasta que se cumpla una de las siguientes condiciones:

- El decodificador de vídeo 30 detecta un inicio de una nueva secuencia de vídeo codificado.
- El decodificador de vídeo 30 detecta un final del flujo de bits de vídeo codificado.
- El decodificador de vídeo 30 determina que el valor del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` señalizado es mayor que cero, y que el bit menos significativo del valor del POC (indicado por `"lsb_recuento_orden_imagen"`) del siguiente fragmento en el orden de decodificación es igual al resultado de la siguiente fórmula: $(\text{LsbActualRecuentoOrdenImagen} + \text{núm_de_etapas_de_refinamiento_menos1} + 1) \% \text{LsbMáximoRecuentoOrdenImagen}$, donde `LsbActualRecuentoOrdenImagen` es el valor de `lsb_recuento_orden_imagen` de la imagen en la unidad de

acceso que contiene el mensaje de SEI. El operador "%" se refiere a una operación de módulo, que produce el valor del resto de una operación de división de enteros.

- El decodificador de vídeo 30 determina que el valor del `núm_de_etapas_de_refinamiento_menos1` señalado es igual a cero, y que se decodifica un mensaje señalado de SEI de fin de segmento de refinamiento progresivo, con el mismo valor de `identificador_de_refinamiento_progresivo` que el del mensaje actual de SEI de inicio de segmento de refinamiento progresivo.

Además, el decodificador de vídeo 30 puede determinar, cuando una unidad de NAL de SEI señalizada que incluye un mensaje de SEI de inicio de segmento de refinamiento progresivo, e incluye un elemento sintáctico `"nuh_reservado_cero_6bits"` con un valor igual a cero, que la unidad de NAL de SEI señalizada precede, en el orden de decodificación, a la primera unidad de NAL de VCL en una AU actual. En la decodificación del mensaje de SEI de inicio de segmento de refinamiento progresivo, el decodificador de vídeo 30 puede determinar que el valor del elemento sintáctico `identificador_de_refinamiento_progresivo` cae dentro de un rango numérico que comienza en 0 y termina en $(2^{32}-2)$, ambos valores inclusive.

Por ejemplo, si el decodificador de vídeo 30 determina que el elemento sintáctico `identificador_de_refinamiento_progresivo` tiene un valor dentro de los márgenes de 0 a 255, ambos valores inclusive, y de 512 a $(2^{31} - 1)$, ambos valores inclusive, el decodificador de vídeo 30 puede utilizar el valor detectado para identificar una operación específica de refinamiento progresivo, asociada al segmento de refinamiento progresivo. En otras palabras, cualquier valor asignado al elemento sintáctico `identificador_de_refinamiento_progresivo` en las gamas descritas anteriormente puede estar asociado a una operación de refinamiento progresivo particular, soportada en el documento HEVC WD9. A su vez, si el elemento sintáctico `identificador_de_refinamiento_progresivo` tiene un valor en uno de los rangos descritos anteriormente, el decodificador de vídeo 30 puede decodificar cada imagen codificada del segmento de refinamiento progresivo usando la operación específica de refinamiento progresivo identificada por el valor del elemento sintáctico `identificador_de_refinamiento_progresivo`.

Por otra parte, los valores del elemento sintáctico `identificador_de_refinamiento_progresivo` que caen dentro de los rangos de 256 a 511, ambos valores inclusive, y de 2^{31} a $(2^{32} - 2)$, ambos valores inclusive, están reservados para uso futuro por la ITU- T y/o la ISO / IEC. Si el decodificador de vídeo 30 detecta que el elemento sintáctico `identificador_de_refinamiento_progresivo` es igual a un valor en una de las gamas reservadas descritas anteriormente, a continuación, el decodificador de vídeo 30 puede no tener en cuenta el elemento sintáctico `identificador_de_refinamiento_progresivo`. En particular, para no tener en cuenta este elemento sintáctico, el decodificador de vídeo 30 puede quitar el elemento sintáctico `identificador_de_refinamiento_progresivo` del flujo recibido de bits de vídeo codificado, y desechar el elemento sintáctico `identificador_de_refinamiento_progresivo`.

Como se ha descrito, el decodificador de vídeo 30 y/o los componentes del mismo pueden configurarse para soportar la ajustabilidad temporal a escala de flujos de bits de vídeo codificado, como según el documento HEVC WD9. Por ejemplo, el decodificador de vídeo 30 puede recibir un sub-flujo de bits que el elemento de red 68 extrae de un flujo completo de bits de vídeo codificado, y comunica al decodificador de vídeo 30. En este ejemplo, el elemento de red 68 puede extraer un subconjunto temporal de imágenes codificadas a partir del conjunto completo de imágenes codificadas incluidas en el flujo completo de bits de vídeo codificado, y proporcionar el subconjunto temporal como parte del sub-flujo de bits al decodificador de vídeo 30. Por ejemplo, el subconjunto temporal puede representar un verdadero subconjunto del conjunto completo de imágenes codificadas incluidas en el flujo completo de bits de vídeo codificado. En escenarios donde el subconjunto temporal representa un verdadero subconjunto del conjunto completo de imágenes codificadas, el flujo completo de bits de vídeo codificado puede incluir datos para cada imagen codificada del subconjunto temporal, y datos para al menos una imagen codificada adicional que no se incluya en el subconjunto temporal.

Para soportar diversas velocidades temporales de imagen, según la ajustabilidad temporal a escala, el decodificador de vídeo 30 puede estar configurado para recibir y decodificar sub-flujos de bits de diferentes velocidades de imagen, tales como diversos sub-flujos de bits que el elemento de red 68 puede extraer del flujo completo de bits de vídeo codificado. Más específicamente, los diferentes sub-flujos de bits que incluyen subconjuntos temporales de recuentos variables de imágenes pueden representar diferentes velocidades de imagen. Para soportar la ajustabilidad temporal a escala, el decodificador de vídeo 30 puede decodificar cualquier sub-flujo de bits, independientemente de la velocidad de imágenes, como un flujo de bits decodificable independientemente. En otras palabras, el decodificador de vídeo 30 puede decodificar un subconjunto temporal particular de imágenes codificadas sin ningún dato adicional, tal como la información incluida en el flujo completo de bits de vídeo codificado, pero excluida del sub-flujo específico de bits.

En ejemplos donde el decodificador de vídeo 30 recibe un flujo completo de bits de vídeo codificado, señalado mediante un dispositivo de codificación de vídeo, el flujo completo de bits de vídeo codificado puede incluir una o más sub-capas temporales. Además, cada unidad de NAL recibida y/o decodificada por el decodificador de vídeo 30 puede pertenecer a una sub-capa específica, según lo indicado por un valor correspondiente de `"IdentificadorTemporal"`. Más específicamente, el decodificador de vídeo 30 puede determinar que el valor del

IdentificadorTemporal de una unidad de NAL es igual al valor del correspondiente elemento sintáctico "identificador_temporal_más1" señalizado, menos uno. Además, el decodificador de vídeo 30 puede determinar que todas las unidades señalizadas de NAL de VCL de una sola imagen pertenecen a una sola sub-capa (es decir, la misma sub-capa). En otras palabras, el decodificador de vídeo 30 puede decodificar una imagen codificada, sobre la base de una determinación de que la propia imagen codificada pertenece a la sub-capa específica correspondiente a las unidades de NAL asociadas a la imagen codificada.

Por ejemplo, según el documento HEVC WD9, el decodificador de vídeo 30 puede decodificar un flujo señalizado de bits de vídeo codificado, de manera que un proceso de decodificación de una sub-capa inferior del flujo de bits no dependa de los datos en una sub-capa superior del flujo de bits. El elemento de red 68 puede generar un sub-flujo de bits a partir de un flujo de bits completo, mediante la eliminación, del flujo de bits completo, de todas las unidades de NAL asociadas a un valor de IdentificadorTemporal que sea superior a un valor específico. El dispositivo de codificación de vídeo puede garantizar que todas las condiciones para la conformidad del flujo de bits (por ejemplo, restricciones de almacenamiento temporal) con respecto al documento HEVC WD9, se cumplen con respecto al flujo de bits completo y, por lo tanto, para cada sub-flujo de bits que el elemento de red 68 pueda extraer del flujo completo de bits. A su vez, el decodificador de vídeo 30 puede decodificar cualquier sub-flujo de bits señalizado sin ningún cambio en el proceso de decodificación, y sin que sea necesario ningún cambio en la infraestructura de hardware y/o software. En otras palabras, el decodificador de vídeo 30 puede decodificar un sub-flujo de bits señalizado, soportando al mismo tiempo la ajustabilidad temporal a escala, según el documento HEVC WD9, de manera correspondiente a la decodificación de un flujo completo de bits de vídeo codificado.

Como se ha descrito, en el ajuste temporal a escala de un flujo completo de bits de vídeo codificado, el elemento de red 68 puede extraer un subconjunto temporal de imágenes codificadas a partir del flujo completo de bits de vídeo codificado. Más específicamente, el subconjunto temporal puede ser un verdadero subconjunto de las imágenes codificadas señalizadas en el flujo completo de bits de vídeo codificado y, por lo tanto, el elemento de red 68 puede eliminar una o más imágenes codificadas del flujo completo de bits codificado para generar el sub-flujo de bits. En los ejemplos, el elemento de red 68 puede eliminar una o más imágenes codificadas que se incluyen en un segmento de refinamiento progresivo. El decodificador de vídeo 30 puede recibir, potencialmente, un valor incorrecto del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1`. Al decodificar y, potencialmente, aplicar un valor incorrecto del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1`, el decodificador de vídeo 30 puede aplicar la operación de refinamiento progresivo para decodificar las imágenes que se sitúan después del límite final del segmento de refinamiento progresivo ajustado temporalmente a escala.

Para mitigar o eliminar la decodificación errónea de las imágenes que se sitúan después del final del segmento de refinamiento progresivo en el flujo de bits de vídeo codificado, el decodificador de vídeo 30 y/o los componentes del mismo, tales como la unidad de decodificación por entropía 70, pueden implementar una o más técnicas de esta divulgación. Según algunas implementaciones de las técnicas, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener información que indica el valor del recuento de orden de imágenes (POC) de la última imagen del segmento de refinamiento progresivo. El valor del POC puede representar un valor estático que se asocia a una imagen codificada en particular, señalizada en el flujo de bits de vídeo codificado y, por lo tanto, puede identificar la imagen codificada en particular al decodificador de vídeo 30, independientemente de cualquier ajuste temporal a escala del flujo de bits de vídeo codificado.

Por ejemplo, al obtener el valor del POC de la última imagen codificada del segmento de refinamiento progresivo, el decodificador de vídeo 30 puede identificar con más precisión la última imagen codificada del segmento de refinamiento progresivo. Más específicamente, el decodificador de vídeo 30 puede identificar la última imagen codificada del segmento de refinamiento progresivo, independientemente de si el decodificador de vídeo 30 ha recibido o no un segmento de refinamiento progresivo ajustado temporalmente a escala. En ejemplos donde el decodificador de vídeo 30 recibe un segmento de refinamiento progresivo ajustado temporalmente a escala, el decodificador de vídeo 30 puede identificar con precisión la última imagen del segmento de refinamiento progresivo, sin tener en cuenta el grado en que el segmento de refinamiento progresivo haya sido ajustado temporalmente a escala. En otras palabras, el decodificador de vídeo 30 puede identificar la última imagen codificada (es decir, el límite final) del segmento de refinamiento progresivo, independientemente de la velocidad de imágenes del flujo recibido de bits de vídeo codificado.

Según diversas implementaciones de las técnicas descritas en el presente documento, el decodificador de vídeo 30 puede decodificar un mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener información que el decodificador de vídeo 30 puede usar para determinar el valor del POC asociado a la última imagen codificada del segmento de refinamiento progresivo. En una implementación, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener datos que indican un valor de "delta de POC". Más específicamente, el valor de delta de POC puede indicar una diferencia entre el valor del POC de la imagen de base y el valor del POC de la última imagen codificada del segmento de refinamiento progresivo.

A su vez, el decodificador de vídeo 30 puede utilizar el mensaje de SEI de inicio de segmento de refinamiento progresivo para detectar el límite inicial (por ejemplo, la primera imagen codificada) de un segmento correspondiente

de refinamiento progresivo. Además, el decodificador de vídeo 30 puede determinar el límite final (por ejemplo, la última imagen codificada) del segmento de refinamiento progresivo, aplicando el valor de delta de POC al valor del POC de la imagen de base (por ejemplo, la imagen codificada que precede inmediatamente al límite inicial, en el orden de decodificación). Por ejemplo, mediante la aplicación del valor de delta de POC al valor del POC de la imagen de base, el decodificador de vídeo 30 puede obtener el valor del POC del límite final del segmento de refinamiento progresivo.

Como se ha descrito, el valor del POC de una imagen codificada en particular puede ser un valor estático asociado a la imagen codificada, y puede indicar la posición de la imagen codificada en el orden de decodificación. Al obtener datos a partir de los cuales obtener el valor del POC del límite final del segmento de refinamiento progresivo, el decodificador de vídeo 30 puede determinar el límite final del segmento de refinamiento progresivo, con independencia de si, y en qué magnitud, el elemento de red 68 ajusta temporalmente a escala, o no, el segmento de refinamiento progresivo. El decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener el valor de delta de POC a partir de un elemento sintáctico "delta_recuento_orden_imagen", tal como se ilustra en la tabla de sintaxis 3 anterior. De esta manera, el decodificador de vídeo 30 puede implementar una o más técnicas de esta divulgación para decodificar una secuencia de refinamiento progresivo utilizando una correspondiente operación de refinamiento progresivo, soportando al mismo tiempo la ajustabilidad temporal a escala de los flujos de bits de vídeo codificado.

Según algunas implementaciones en las que el decodificador de vídeo 30 decodifica el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener los datos asociados al valor del POC del límite final del segmento de refinamiento progresivo, el decodificador de vídeo 30 puede obtener un delta de los bits menos significativos (LSB) del mensaje decodificado de SEI de inicio de segmento de refinamiento progresivo. En estas implementaciones, el decodificador de vídeo 30 puede utilizar el delta de los LSB para determinar una diferencia entre el valor de los LSB del valor del POC de la imagen de base, y el valor de los LSB del valor del POC de la última imagen codificada del segmento de refinamiento progresivo.

A su vez, el decodificador de vídeo 30 puede utilizar el delta de los LSB para obtener el valor del POC del límite final del segmento de refinamiento progresivo. Más específicamente, el decodificador de vídeo 30 puede aplicar el delta de los LSB a los LSB del valor del POC de la imagen de base. En un ejemplo, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener un elemento sintáctico que indica el delta de los LSB, en lugar del elemento sintáctico "núm_de_etapas_de_refinamiento_menos1".

Con la obtención del elemento sintáctico que indica el delta de los LSB, el decodificador de vídeo 30 puede determinar el límite final del segmento de refinamiento progresivo, con independencia de si, y en qué magnitud, el elemento de red 68 ajusta temporalmente a escala, o no, el segmento de refinamiento progresivo. Por ejemplo, el decodificador de vídeo 30 puede aplicar el valor del elemento sintáctico decodificado para obtener el valor del POC del límite final del segmento de refinamiento progresivo. El decodificador de vídeo 30 puede utilizar el valor obtenido del POC para identificar el límite final, utilizando un valor estático asociado al límite final, al decodificar el segmento de refinamiento progresivo. De esta manera, el decodificador de vídeo 30 puede implementar una o más técnicas de esta divulgación para decodificar una secuencia de refinamiento progresivo utilizando una correspondiente operación de refinamiento progresivo, soportando al mismo tiempo la ajustabilidad temporal a escala de los flujos de bits de vídeo codificado.

Según otras implementaciones de las técnicas descritas en el presente documento, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener datos que indican el número de imágenes codificadas consecutivas que 1) están asociados a un valor de IdentificadorTemporal que es menor o igual al valor de IdentificadorTemporal de la imagen base y 2) representan refinamientos progresivos de calidad de la imagen de base. Más específicamente, un valor decreciente de IdentificadorTemporal puede indicar un avance en el flujo de bits de vídeo codificado, según el orden de decodificación. Por ejemplo, la primera imagen codificada del segmento de refinamiento progresivo puede tener un valor de IdentificadorTemporal que es menor que el valor de IdentificadorTemporal de la imagen de base. A su vez, la segunda imagen codificada del segmento de refinamiento progresivo puede tener un valor de IdentificadorTemporal que es menor que el IdentificadorTemporal de la primera imagen codificada del segmento de refinamiento progresivo, y así sucesivamente.

La imagen de base puede ser la imagen codificada incluida en la unidad de acceso actual y la imagen de base puede preceder inmediatamente a la primera imagen codificada del segmento de refinamiento progresivo, en el orden de decodificación. En algunos ejemplos, la imagen de base también puede preceder inmediatamente a la primera imagen codificada del segmento de refinamiento progresivo, en el orden de salida. En algunos de tales ejemplos, el segmento de refinamiento progresivo puede incluir una secuencia de imágenes codificadas que son consecutivas, tanto en el orden de decodificación como en el orden de salida.

Según estas implementaciones de las técnicas descritas en el presente documento, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener datos que indican el número de imágenes codificadas consecutivas que tienen un valor de IdentificadorTemporal menor que el

valor de IdentificadorTemporal de la imagen de base, así como representar refinamientos progresivos de calidad de la imagen de base. Por ejemplo, la información obtenida por el decodificador de vídeo 30 puede indicar el número de imágenes codificadas en la secuencia que conforma el segmento de refinamiento progresivo. En estas implementaciones, el decodificador de vídeo 30 puede recibir, desde el elemento de red 68, una serie de imágenes codificadas consecutivas que cumplan las condiciones enumeradas anteriormente. Más específicamente, el elemento de red 68 puede determinar el número de imágenes codificadas consecutivas que cumplen las condiciones, por separado, para cada subconjunto temporal de imágenes codificadas que el elemento de red 68 extrae desde el flujo completo de bits de vídeo codificado. A su vez, el elemento de red 68 puede comunicar el número de imágenes codificadas consecutivas que cumplen las condiciones enumeradas anteriormente al decodificador de vídeo 30.

Mediante la aplicación del número recibido de imágenes codificadas consecutivas que satisfacen las condiciones anteriores, el decodificador de vídeo 30 puede determinar con mayor precisión la longitud del segmento de refinamiento progresivo, incluso en escenarios en los que el elemento de red 68 ajusta temporalmente a escala el segmento de refinamiento progresivo. A su vez, el decodificador de vídeo 30 puede obtener el número de imágenes codificadas consecutivas que satisfacen las condiciones como parte del flujo de bits de vídeo codificado, tanto si el flujo de bits de vídeo codificado representa, como si no, un sub-flujo de bits (por ejemplo, representando un subconjunto temporal de las imágenes codificadas). En diversos ejemplos, el decodificador de vídeo 30 puede generar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener un elemento sintáctico que indica el número determinado de imágenes codificadas consecutivas que satisfagan las condiciones anteriores. Por ejemplo, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener el elemento sintáctico que indica el número determinado, en lugar del elemento sintáctico `núm_de_etapas_de_refinamiento_menos1` ilustrado en la tabla de sintaxis 1 anteriormente.

El decodificador de vídeo 30 puede, mediante la obtención del número de imágenes codificadas consecutivas que tienen un valor de IdentificadorTemporal menor que el valor de IdentificadorTemporal de la imagen de base y que representan refinamientos progresivos de calidad de la imagen de base, determinar con mayor precisión la longitud del segmento de refinamiento progresivo. Además, mediante la obtención de la información descrita anteriormente, como parte de la decodificación de un mensaje codificado de SEI de inicio de segmento de refinamiento progresivo, el decodificador de vídeo 30 puede determinar con precisión la longitud del segmento de refinamiento progresivo, independientemente de si el elemento de red 68 ajusta temporalmente a escala, o no, el segmento de refinamiento progresivo. Por ejemplo, el decodificador de vídeo 30 puede utilizar el elemento sintáctico obtenido del mensaje de SEI de inicio de segmento de refinamiento progresivo para determinar la longitud del segmento de refinamiento progresivo, independientemente de si, o en qué magnitud, el elemento de red 68 ajusta temporalmente a escala, o no, el segmento de refinamiento progresivo. De esta manera, el decodificador de vídeo 30 puede implementar una o más técnicas de esta divulgación para determinar una serie de imágenes codificadas consecutivas que 1) tienen un valor de IdentificadorTemporal menor que el IdentificadorTemporal de la imagen de base y 2) representan refinamientos progresivos de calidad de la imagen de base, permitiendo de este modo a un dispositivo de decodificación de vídeo decodificar un segmento de refinamiento progresivo, soportando al mismo tiempo los flujos de bits de vídeo temporalmente ajustables a escala.

Como se ha descrito anteriormente con respecto a la FIG. 3, el decodificador de vídeo 30 y/o los componentes del mismo pueden realizar un procedimiento de decodificación de datos de vídeo, incluyendo el procedimiento la recepción de información que indica un valor del recuento de orden de imagen (POC) de una última imagen entre una pluralidad de imágenes en un segmento de refinamiento progresivo en un flujo de bits codificado, y la decodificación de al menos algunas de las imágenes en el segmento de refinamiento progresivo, según una operación de refinamiento progresivo basada en la información recibida. En algunos ejemplos de implementaciones del procedimiento descrito anteriormente con respecto al decodificador de vídeo 30, la información recibida incluye un valor de delta de POC que indica una diferencia entre un valor del POC de una imagen que precede inmediatamente al segmento de refinamiento progresivo y el valor del POC de la última imagen del segmento de refinamiento progresivo. Según algunos ejemplos de implementaciones del procedimiento descrito anteriormente con respecto al decodificador de vídeo 30, la información recibida incluye un delta de los bits menos significativos (LSB), que indica una diferencia entre los LSB de un valor del POC de una imagen que precede inmediatamente al segmento de refinamiento progresivo y los LSB de un valor del POC de la última imagen del segmento de refinamiento progresivo. En un ejemplo del procedimiento descrito anteriormente con respecto al decodificador de vídeo 30, la información recibida incluye un valor del POC de la última imagen del segmento de refinamiento progresivo.

Según algunas implementaciones del procedimiento descrito anteriormente con respecto al decodificador de vídeo 30, la recepción de la información comprende recibir la información en un mensaje de información de mejora suplementaria (SEI) incluido en el flujo de bits codificado. En una implementación de ese tipo, el mensaje de SEI comprende un mensaje de SEI de inicio de segmento de refinamiento progresivo que indica un límite inicial del segmento de refinamiento progresivo. Según algunas implementaciones del procedimiento descrito anteriormente con respecto al decodificador de vídeo 30, el procedimiento puede incluir además la determinación de que una imagen codificada precede inmediatamente al segmento de refinamiento progresivo en un flujo de bits de vídeo codificado. En una de tales implementaciones, cada una entre la pluralidad de imágenes en la secuencia de

refinamiento progresivo define un refinamiento de la calidad en comparación con la imagen codificada que precede inmediatamente a la secuencia de refinamiento progresivo.

En algunos ejemplos del procedimiento descrito anteriormente con respecto al decodificador de vídeo 30, el procedimiento puede comprender además detectar la última imagen entre la pluralidad de imágenes en el segmento de refinamiento progresivo, utilizando la información recibida, y dejar de decodificar según la operación de refinamiento progresivo. Según algunas implementaciones del procedimiento descrito anteriormente con respecto al decodificador de vídeo 30, la decodificación de al menos algunas de las imágenes en el segmento de refinamiento progresivo, según la operación de refinamiento progresivo basada en la información recibida, puede comprender la aplicación de la operación de refinamiento progresivo a cada imagen en el segmento de refinamiento progresivo, a excepción de la última imagen en el segmento de refinamiento progresivo.

En diversos ejemplos, el decodificador de vídeo 30 puede estar incluido en un dispositivo para codificar datos de vídeo, tal como un ordenador de sobremesa, un ordenador plegable (es decir, portátil), un ordenador tipo tablet, un decodificador, un equipo telefónico portátil tal como un denominado teléfono "inteligente", un denominado panel "inteligente", un televisor, una cámara, un dispositivo de visualización, un reproductor de medios digitales, una consola de videojuegos, un dispositivo de flujos de transmisión de vídeo o similares. En los ejemplos, un dispositivo de este tipo para la codificación de datos de vídeo puede incluir uno o más entre un circuito integrado, un microprocesador y un dispositivo de comunicación que incluye el decodificador de vídeo 30.

La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de segmento de refinamiento progresivo 94 y la correspondiente imagen de base 92, según uno o más aspectos de esta divulgación. Más específicamente, la FIG. 4 ilustra la secuencia de imágenes 90, que incluye la imagen de base 92 y el segmento de refinamiento progresivo 94. A su vez, el segmento de refinamiento progresivo 94 incluye las imágenes de refinamiento progresivo 94A a 94N. Cada una de las imágenes de refinamiento progresivo 94A a 94N puede representar un refinamiento de calidad de la imagen de base 92. En varios ejemplos, un refinamiento de calidad puede, según esta divulgación, representar un refinamiento en términos de reducir progresivamente una magnitud de distorsión entre una imagen codificada y una imagen original. Por ejemplo, cada una de las imágenes de refinamiento progresivo 94A a 94N puede incluir datos residuales que la imagen de base 92 no incluye. Además, cada una de las imágenes de refinamiento progresivo 94A a 94N puede representar refinamientos de calidad de la imagen de base 92, sin representar ninguna predicción basada en el movimiento a partir de la imagen de base 92.

Más específicamente, el segmento de refinamiento progresivo 94 puede representar una secuencia de imágenes codificadas en orden de decodificación consecutiva. En algunos ejemplos, la secuencia de imágenes codificadas incluidas en el segmento de refinamiento progresivo 94 también puede estar en orden de salida consecutiva. Además, el segmento de refinamiento progresivo 94 puede representar un refinamiento continuo de la calidad con respecto a la imagen de base 92, definido por un incremento por imagen. Por ejemplo, la imagen de refinamiento progresivo 94A, que es la primera imagen del segmento de refinamiento progresivo 94, puede representar un refinamiento de calidad con respecto a la imagen de base 82, en un incremento predefinido. A su vez, la imagen de refinamiento progresivo 94B, que es la segunda imagen del segmento de refinamiento progresivo 94, puede representar un refinamiento de calidad con respecto a la imagen de refinamiento progresivo 94A, en el incremento predefinido, y así sucesivamente. De esta manera, el segmento de refinamiento progresivo 94 puede representar un refinamiento iterativo de la calidad con respecto a la imagen de base 92, con la última imagen codificada (imagen de refinamiento progresivo 94N) representando el refinamiento acumulativo de calidad entre todas las iteraciones del incremento de refinamiento predefinido.

Por ejemplo, el decodificador de vídeo 30 puede decodificar el segmento de refinamiento progresivo 94 mediante la aplicación de una operación de refinamiento progresivo a la imagen anterior en el orden de decodificación. Más específicamente, el decodificador de vídeo 30 puede recibir un mensaje de SEI de inicio de segmento de refinamiento progresivo como parte de un flujo de bits de vídeo codificado, señalado por el codificador de vídeo 20. Además, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener un valor de un elemento sintáctico `identificador_de_refinamiento_progresivo`. A su vez, el decodificador de vídeo 30 puede utilizar el valor obtenido del `identificador_de_refinamiento_progresivo` para determinar la operación específica de refinamiento progresivo con la cual decodificar el segmento de refinamiento progresivo 94. El decodificador de vídeo 30 puede aplicar la operación determinada de refinamiento progresivo a la imagen de base 92, y a cada imagen del segmento de refinamiento progresivo 94, a excepción de la imagen de refinamiento progresivo 94N (es decir, la última imagen). Más específicamente, el decodificador de vídeo 30 puede aplicar la operación de refinamiento progresivo, sin aplicar ninguna predicción de movimiento, a la imagen de base 92 y a cada imagen del segmento de refinamiento progresivo 94, hasta la penúltima imagen.

Además, el decodificador de vídeo 30 puede implementar una o más técnicas de esta divulgación para decodificar el segmento de refinamiento progresivo 94, soportando al mismo tiempo la ajustabilidad temporal a escala de los flujos de bits de vídeo codificado. Por ejemplo, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener información que indique un valor del recuento de orden de imágenes (POC) de la última imagen (por ejemplo, la imagen de refinamiento progresivo 94N) del segmento de refinamiento progresivo 94. El mensaje de SEI de inicio de segmento de refinamiento progresivo puede estar situado

inmediatamente después de los datos codificados asociados a la imagen de base 92, en el flujo de bits de vídeo codificado. El valor del POC de una imagen en particular puede indicar la posición de la imagen en particular dentro del flujo de bits de vídeo codificado, independientemente de si el flujo de bits de vídeo codificado está ajustado temporalmente a escala, o no, para proporcionar una velocidad de imagen reducida. Por ejemplo, el decodificador de vídeo 30 puede identificar una imagen codificada por el valor del POC asignado de la imagen, sin depender de la información asociada a otras partes de un flujo recibido de bits de vídeo codificado.

En algunos ejemplos, el decodificador de vídeo 30 puede decodificar un mensaje de SEI de inicio de segmento de refinamiento progresivo, asociado al segmento de refinamiento progresivo 94, para obtener información que indica el valor del POC de la imagen de refinamiento progresivo 94N. Por ejemplo, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener un valor de un elemento sintáctico incluido en el mensaje de SEI. Más específicamente, en este caso, el decodificador de vídeo 30 puede obtener, a partir del elemento sintáctico decodificado, el valor del POC asociado a la imagen de refinamiento progresivo 94N.

A su vez, el decodificador de vídeo 30 puede decodificar el segmento de refinamiento progresivo 94 por la aplicación iterativa de la operación de refinamiento progresivo identificada por el valor del elemento sintáctico `identificador_de_refinamiento_progresivo`, hasta que el decodificador de vídeo 30 detecte el valor del POC identificado por el elemento sintáctico correspondiente. Al detectar el valor del POC identificado por el elemento sintáctico correspondiente del mensaje de SEI de inicio de segmento de refinamiento progresivo, el decodificador de vídeo 30 puede determinar que el decodificador de vídeo 30 ha alcanzado el límite final del segmento de refinamiento progresivo 94 (que, en el ejemplo de la FIG. 4, está indicado por la imagen de refinamiento progresivo 94N). Sobre la base de la determinación de que el decodificador de vídeo 30 ha alcanzado el límite final del segmento de refinamiento progresivo 94, el decodificador de vídeo 30 puede cesar la decodificación de imágenes codificadas del flujo de bits, en base, exclusivamente, a la operación de refinamiento indicada por el elemento sintáctico `identificador_de_refinamiento_progresivo`.

Según otras implementaciones de las técnicas de esta divulgación, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener un elemento sintáctico `delta_POC`. El decodificador de vídeo 30 puede utilizar el valor del elemento sintáctico `delta_POC` para determinar el valor del POC del límite final de la imagen de refinamiento progresivo 94N, que denota el límite final del segmento de refinamiento progresivo 94. Más específicamente, el decodificador de vídeo 30 puede aplicar el valor de `delta_POC` al valor del POC de la imagen de base 92, para determinar el valor del POC de la imagen de refinamiento progresivo 94N. El decodificador de vídeo 30 puede, a su vez, utilizar el valor determinado del POC de la imagen de refinamiento progresivo 94N para detectar cuándo, en el proceso de decodificación, el decodificador de vídeo 30 ha alcanzado el límite final del segmento de refinamiento progresivo 94. Sobre la base de la determinación de que el decodificador de vídeo 30 ha alcanzado el límite final del segmento de refinamiento progresivo 94, utilizando el valor `delta_POC`, el decodificador de vídeo 30 puede dejar de decodificar según el refinamiento progresivo.

En otros ejemplos de implementaciones de las técnicas descritas en el presente documento, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener un elemento sintáctico que indica un delta de los bits menos significativos (LSB). El decodificador de vídeo 30 puede utilizar el delta del valor de los LSB para detectar la imagen de refinamiento progresivo 94N, durante el proceso de decodificación con respecto al flujo de bits de vídeo codificado. Más específicamente, el decodificador de vídeo 30 puede aplicar el delta del valor de los LSB para los LSB del valor del POC de la imagen de base 92, para obtener los LSB del valor del POC de la imagen de refinamiento progresivo 94N, que conforma el límite final del segmento de refinamiento progresivo 94.

A su vez, el decodificador de vídeo 30 puede decodificar la imagen de refinamiento progresivo 94A únicamente mediante la aplicación de la operación de refinamiento identificada por el elemento sintáctico `identificador_de_refinamiento_progresivo` a la imagen de base 92, y continuar decodificando las imágenes posteriores en el orden de decodificación, continuando la aplicación de la operación de refinamiento, hasta que el decodificador de vídeo 30 identifique una imagen asociada a un valor del POC que tiene el valor obtenido de los LSB. En un ejemplo, el decodificador de vídeo 30 puede decodificar cada una de las imágenes de refinamiento progresivo 94A a 94N aplicando de forma iterativa la operación de refinamiento a la imagen inmediatamente anterior del segmento de refinamiento progresivo 94. En otro ejemplo, el decodificador de vídeo 30 puede decodificar cada una de las imágenes de refinamiento progresivo 94A a 94N mediante la aplicación de la operación de refinamiento progresivo, o una variante de la misma, a la imagen de base 92. Además, al detectar una imagen que tiene un valor del POC con los LSB obtenidos, el decodificador de vídeo 30 puede determinar que el decodificador de vídeo 30 ha alcanzado el límite final del segmento de refinamiento progresivo 94 (es decir, la imagen de refinamiento progresivo 94N) en el proceso de decodificación del flujo de bits de vídeo codificado. Sobre la base de la determinación de que el decodificador de vídeo 30 ha alcanzado el límite final del segmento de refinamiento progresivo 94, utilizando el valor de `delta_POC`, el decodificador de vídeo 30 puede determinar que el decodificador de vídeo 30 ha completado la decodificación del segmento de refinamiento progresivo.

La FIG. 5 es un diagrama de flujo que ilustra un ejemplo de proceso 100, que el decodificador de vídeo 30 y/o los

componentes del mismo pueden realizar para decodificar los datos de vídeo codificado, según uno o más aspectos de esta divulgación. El proceso 100 puede comenzar cuando el decodificador de vídeo 30 identifica una siguiente imagen codificada señalizada en un flujo de bits de vídeo codificado (102). Más específicamente, el decodificador de vídeo 30 puede detectar la siguiente imagen codificada según el orden de decodificación y, en algunos ejemplos, según el orden de salida, señalizado en el flujo de bits de vídeo codificado.

Además, el decodificador de vídeo 30 puede determinar si la imagen codificada identificada precede inmediatamente, o no, a un segmento de refinamiento progresivo en el orden de decodificación (104). Más específicamente, mediante la determinación de si la imagen codificada precede inmediatamente o no a un segmento de refinamiento progresivo, el decodificador de vídeo 30 puede determinar si la imagen codificada forma o no una imagen de base para el segmento de refinamiento progresivo. Por ejemplo, el decodificador de vídeo 30 puede determinar si la imagen codificada es o no una imagen de base, mediante la decodificación por entropía de la imagen codificada, y determinar si un mensaje de SEI de inicio de segmento de refinamiento progresivo se sitúa o no inmediatamente después de la imagen codificada en el flujo de bits de vídeo codificado. Si el decodificador de vídeo 30 detecta el mensaje de SEI de inicio de segmento de refinamiento progresivo situado inmediatamente después de la imagen codificada, el decodificador de vídeo 30 puede determinar que la imagen codificada forma la imagen de base para un segmento de refinamiento progresivo.

Si el decodificador de vídeo 30 determina que la imagen más recientemente decodificada no es una imagen de base para un segmento de refinamiento progresivo (rama "NO" de 104), el decodificador de vídeo 30 puede identificar la siguiente imagen codificada en el orden de decodificación señalizado en el flujo de bits de vídeo codificado (volviendo efectivamente a 102), y continuar con el proceso de decodificación. Por otro lado, si el decodificador de vídeo 30 determina que la imagen más recientemente decodificada es la imagen de base para un segmento de refinamiento progresivo (rama "SÍ" de 104), entonces el decodificador de vídeo 30 puede detectar datos que indican el valor del POC de la última imagen codificada del segmento de refinamiento progresivo (106).

En diversas implementaciones, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener los datos que indican el valor del POC de la última imagen codificada del segmento de refinamiento progresivo. Por ejemplo, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener uno o más elementos sintácticos que indican el valor del POC de la última imagen codificada del segmento de refinamiento progresivo. En diversos ejemplos, el decodificador de vídeo 30 puede utilizar el(los) elemento (s) sintáctico(s) para determinar el valor del POC, ya sea directa o indirectamente. Más específicamente, el(los) elemento (s) sintáctico(s) puede(n) indicar directamente el valor del POC al decodificador de vídeo 30, o puede(n) proporcionar al decodificador de vídeo 30 datos con los cuales obtener el valor del POC de la última imagen codificada en el segmento de refinamiento progresivo. Como ejemplo, el decodificador de vídeo 30 puede decodificar uno o más elementos sintácticos que indican uno o más valores de delta de POC, que indican una diferencia entre los valores del POC de la imagen de base y de la última imagen codificada del segmento de refinamiento progresivo. Como otro ejemplo, el decodificador de vídeo 30 puede decodificar uno o más elementos sintácticos que indican un delta de los LSB, que indica una diferencia entre los LSB de los valores del POC de la imagen de base y de la última imagen codificada del segmento de refinamiento progresivo.

Además, el decodificador de vídeo 30 puede determinar una operación de refinamiento progresivo asociada al segmento de refinamiento progresivo (108). Más específicamente, el decodificador de vídeo 30 puede decodificar el mensaje de SEI de inicio de segmento de refinamiento progresivo para obtener un elemento sintáctico `identificador_de_refinamiento_progresivo`. A su vez, el decodificador de vídeo 30 puede utilizar el valor del elemento sintáctico `identificador_de_refinamiento_progresivo` para identificar una operación de refinamiento progresivo, que se utiliza para decodificar el segmento de refinamiento progresivo. Por ejemplo, el decodificador de vídeo 30 puede, basándose en el valor del elemento sintáctico `identificador_de_refinamiento_progresivo`, decodificar una o más de las imágenes de refinamiento progresivo 94A a 94N mediante la adición de los datos residuales a los datos de vídeo decodificados asociados a la imagen de base 92.

El decodificador de vídeo 30 puede decodificar el segmento de refinamiento progresivo, utilizando los datos que indican el valor del POC de la última imagen del segmento de refinamiento progresivo, y la operación de refinamiento progresivo identificada (110). Más específicamente, el decodificador de vídeo 30 puede aplicar iterativamente la operación identificada de refinamiento progresivo a la imagen de base, y a cada siguiente imagen en el orden de decodificación, hasta que el decodificador de vídeo 30 detecte el valor del POC (o los LSB del mismo) asociado a la última imagen codificada del refinamiento progresivo segmento. Por ejemplo, para aplicar de forma iterativa la operación de refinamiento progresivo, el decodificador de vídeo 30 puede determinar datos adicionales a partir de cada sucesiva imagen codificada del segmento, y usar los datos adicionales para añadirlos a una profundidad de bits de una imagen dada del segmento. El POC puede ser un valor estático que no se ve afectado por el ajuste temporal a escala del flujo de bits de vídeo codificado. Como resultado, el decodificador de vídeo 30 puede utilizar los datos indicadores del POC y la operación identificada de refinamiento progresivo para decodificar el segmento de refinamiento progresivo según una operación de refinamiento progresivo como se define en el documento HEVC WD9, soportando al mismo tiempo los flujos de bits de vídeo temporalmente ajustables a escala.

La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de proceso 120, que el codificador de vídeo 20 y/o los componentes del mismo pueden realizar para codificar datos de vídeo, según uno o más aspectos de esta divulgación. El proceso 120 puede comenzar cuando el codificador de vídeo 20 forma una siguiente imagen codificada de un flujo de bits (122). Por ejemplo, el codificador de vídeo 20 puede codificar por entropía la siguiente imagen de los datos de vídeo recibidos, para formar la siguiente imagen codificada. Además, el codificador de vídeo 20 puede determinar si la imagen más recientemente codificada precede inmediatamente a un segmento de refinamiento progresivo en el orden de decodificación (124).

En otras palabras, el codificador de vídeo 20 puede determinar si la imagen más recientemente codificada es una imagen de base para el segmento de refinamiento progresivo. El codificador de vídeo 20 puede determinar si se inserta o no un mensaje de SEI de inicio de segmento de refinamiento progresivo antes de una secuencia de imágenes que sigue inmediatamente a la imagen más recientemente codificada en el orden de decodificación, y definir refinamientos de calidad con respecto a la imagen más recientemente codificada, sin definir ninguna predicción de movimiento con respecto a la imagen más recientemente codificada. Si el codificador de vídeo 20 determina que la imagen más recientemente codificada no es una imagen de base para un segmento de refinamiento progresivo (rama "NO" de 124), el codificador de vídeo 20 puede formar la siguiente imagen codificada que va a señalizarse en el flujo de bits de vídeo codificado.

Por otro lado, si el codificador de vídeo 20 determina que la imagen más recientemente codificada es la imagen de base de un segmento de refinamiento progresivo, el codificador de vídeo 20 puede codificar datos que indican un valor del POC de la última imagen codificada del segmento de refinamiento progresivo (126). Por ejemplo, el codificador de vídeo 20 puede implementar técnicas de esta divulgación para codificar un mensaje de SEI de inicio de segmento de refinamiento progresivo que incluye información con la cual un dispositivo de decodificación de vídeo puede determinar el valor del POC de la última imagen codificada del segmento de refinamiento progresivo. En un ejemplo, el codificador de vídeo 20 puede incluir datos que indican directamente el valor del POC de la última imagen codificada del segmento de refinamiento progresivo, como en la forma de un elemento sintáctico. En otros ejemplos, el codificador de vídeo 20 puede incluir datos que el dispositivo de decodificación de vídeo puede utilizar para obtener el valor del POC.

En algunos ejemplos, el codificador de vídeo 20 puede incluir un elemento sintáctico `delta_recuento_orden_imagen` en el mensaje de SEI de inicio de segmento de refinamiento progresivo. Por ejemplo, el codificador de vídeo 20 puede generar el elemento sintáctico `delta_recuento_orden_imagen` para indicar una diferencia entre los respectivos valores del POC de la imagen de base y de la última imagen codificada del segmento de refinamiento progresivo. Como otro ejemplo, el codificador de vídeo 20 puede incluir un elemento sintáctico que indica un delta de los LSB en el mensaje de SEI. En este ejemplo, el codificador de vídeo 20 puede generar el delta de los LSB para indicar una diferencia entre los respectivos LSB de los valores del POC de la imagen de base y de la última imagen codificada del segmento de refinamiento progresivo. El codificador de vídeo 20 puede habilitar un dispositivo de decodificación de vídeo para detectar el límite final (es decir, la última imagen codificada) del segmento de refinamiento progresivo, utilizando los valores de los elementos sintácticos `delta_recuento_orden_imagen` y/o el delta de los LSB.

Además, el codificador de vídeo 20 puede codificar el segmento de refinamiento progresivo en base a los datos que indican el valor del POC de la última imagen codificada del segmento de refinamiento progresivo de datos, y a una operación de refinamiento progresivo asociada al segmento de refinamiento progresivo. Por ejemplo, el codificador de vídeo 20 puede codificar cada imagen del segmento de refinamiento progresivo mediante la aplicación de la operación de refinamiento progresivo a la imagen codificada inmediatamente anterior. El codificador de vídeo 20 puede detectar el límite final del segmento de refinamiento progresivo en base a la detección del valor del POC (o los LSB del mismo) de la última imagen codificada del segmento de refinamiento progresivo. Sobre la base de la detección del límite final del segmento de refinamiento progresivo, el codificador de vídeo 20 puede dejar de codificar las imágenes subsiguientes utilizando exclusivamente la operación de refinamiento progresivo.

El codificador de vídeo 20 puede señalizar el flujo de bits de vídeo codificado. Más específicamente, el codificador de vídeo 20 puede señalizar el flujo de bits de vídeo codificado para incluir datos asociados a la imagen de base, el mensaje de SEI de inicio de segmento de refinamiento progresivo que incluye uno o más de los elementos sintácticos descritos anteriormente, y el segmento de refinamiento progresivo. En algunos ejemplos, un dispositivo intermedio puede extraer un sub-flujo de bits, por ejemplo ajustando temporalmente a escala un flujo completo de bits de vídeo codificado. En estos ejemplos, debido a que el codificador de vídeo 20 genera y señaliza el mensaje de SEI de inicio de segmento de refinamiento progresivo para incluir datos que indican el valor del POC del límite final del segmento de refinamiento progresivo, el codificador de vídeo 20 puede proporcionar a un dispositivo de decodificación de vídeo datos exactos, por los cuales decodificar el segmento, con independencia del ajuste temporal a escala. El codificador de vídeo 20 puede implementar técnicas de esta divulgación para codificar, y permitir la decodificación de, los segmentos de refinamiento progresivo, soportando al mismo tiempo la ajustabilidad temporal a escala de los flujos de bits de vídeo codificado.

De esta manera, bien el decodificador de vídeo 30, o bien el codificador de vídeo 20, puede ser un ejemplo de un dispositivo para la codificación de datos de vídeo, incluyendo el dispositivo medios para determinar información que indica un valor del recuento de orden de imagen (POC) de una última imagen entre una pluralidad de imágenes en

un segmento de refinamiento progresivo en un flujo de bits codificado, y medios para la codificación de al menos algunas de las imágenes en el segmento de refinamiento progresivo, según una operación de refinamiento progresivo.

5 Además, de esta manera, bien el dispositivo de destino 14, o bien el dispositivo de origen 12, puede ser un ejemplo de un dispositivo informático que incluye, o está acoplado a, un medio de almacenamiento legible por ordenador que tiene almacenadas en el mismo instrucciones que, al ejecutarse, hacen que un procesador programable del dispositivo informático determine la información que indica un valor del recuento de orden de imágenes (POC) de una última imagen entre una pluralidad de imágenes en un segmento de refinamiento progresivo en un flujo de bits
10 codificado, y codifique al menos algunas de las imágenes en el segmento de refinamiento progresivo según una operación de refinamiento progresivo.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones, como una o más instrucciones o código,
15 pueden almacenarse en, o transmitirse por, un medio legible por ordenador, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos, diversos dispositivos de almacenamiento legibles por ordenador, o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, según un
20 protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, generalmente, a (1) medios de almacenamiento tangibles y legibles por ordenador, que sean no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera, a los que se pueda 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.
25

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
30 usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión puede denominarse adecuadamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota 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
35 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. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco de láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente reproducen datos de manera magnética, mientras que otros discos reproducen los datos de manera óptica con láser. Las combinaciones de lo que antecede también deberían incluirse dentro del alcance de los medios legibles por ordenador.
40

45 Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones lógicas programables en 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 cualquier estructura anterior 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 hardware especializado y/o módulos de software configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.
50

55 Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un equipo portátil inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Varios 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 la realización mediante diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, pueden combinarse diversas unidades en una unidad de hardware de códec o ser proporcionadas por una colección de
60 unidades de hardware inter-operativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, junto con el software y/o firmware adecuado.

REIVINDICACIONES

1. Un procedimiento de decodificación de datos de vídeo, procedimiento que comprende:
 - 5 recibir, en un mensaje de información de mejora suplementaria (SEI), un valor delta del recuento de orden de imágenes delta, POC, valor que indica una diferencia entre un valor del POC de una imagen que precede inmediatamente a una pluralidad de imágenes en un segmento de refinamiento progresivo en un flujo de bits de vídeo codificado, y un valor del POC de una última imagen entre la pluralidad de imágenes en el segmento de refinamiento progresivo; y
 - 10 en base al valor delta de POC recibido, decodificar al menos algunas de las imágenes en el segmento de refinamiento progresivo según una operación de refinamiento progresivo, en el que la operación de refinamiento progresivo indica un refinamiento con respecto a la calidad de una imagen.
2. El procedimiento de la reivindicación 1, en el que el mensaje de SEI se compone de un mensaje de SEI de inicio de segmento de refinamiento progresivo que indica un límite inicial del segmento de refinamiento progresivo.
3. El procedimiento de la reivindicación 1, que comprende además:
 - 20 la determinación de que la imagen que precede inmediatamente al segmento de refinamiento progresivo en el flujo de bits de vídeo codificado comprende una imagen de base con respecto al segmento de refinamiento progresivo.
4. El procedimiento de la reivindicación 3, en el que cada imagen respectiva entre la pluralidad de imágenes en el segmento de refinamiento progresivo define un refinamiento de la calidad en comparación con una imagen que precede inmediatamente a la imagen respectiva en el segmento de refinamiento progresivo.
5. El procedimiento de la reivindicación 1, que comprende además:
 - 30 la detección de la última imagen entre la pluralidad de imágenes en el segmento de refinamiento progresivo, usando el valor de delta del POC recibido; y el cese de la decodificación según la operación de refinamiento progresivo.
6. El procedimiento de la reivindicación 1, en el que la decodificación de al menos algunas de las imágenes en el segmento de refinamiento progresivo según la operación de refinamiento progresivo basada en el valor de delta del POC recibido comprende:
 - 40 la aplicación de la operación de refinamiento progresivo a cada imagen en el segmento de refinamiento progresivo, a excepción de la última imagen en el segmento de refinamiento progresivo.
7. Un procedimiento de codificación de datos de vídeo temporalmente ajustables a escala, procedimiento que comprende:
 - 45 la codificación, según una operación de refinamiento progresivo, de al menos algunas imágenes entre una pluralidad de imágenes en un segmento de refinamiento progresivo, en el que la operación de refinamiento progresivo indica un refinamiento con respecto a la calidad de una imagen;
 - 50 la generación de un valor delta del recuento del orden de imágenes, POC, que indica una diferencia entre un valor del POC de una imagen que precede inmediatamente al segmento de refinamiento progresivo en un flujo de bits codificado y el valor del POC de una última imagen del segmento de refinamiento progresivo en el flujo de bits codificados; y
 - la inclusión, en un mensaje de SEI de información de mejora suplementaria, a ser señalado en el flujo de bits de vídeo codificado, del valor generado del delta del POC.
8. El procedimiento de la reivindicación 7, en el que el mensaje de SEI se compone de un mensaje de SEI de inicio de segmento de refinamiento progresivo que indica un límite inicial del segmento de refinamiento progresivo.
9. El procedimiento de la reivindicación 7, que comprende además:
 - 60 la determinación de que la imagen que precede inmediatamente al segmento de refinamiento progresivo en el flujo de bits de vídeo codificado comprende una imagen de base con respecto al segmento de refinamiento progresivo.
10. El procedimiento de la reivindicación 9, en el que cada imagen respectiva entre la pluralidad de imágenes en el segmento de refinamiento progresivo define un refinamiento de la calidad en comparación con una imagen que precede inmediatamente a la imagen respectiva en el segmento de refinamiento progresivo.

11. El procedimiento de la reivindicación 7, que comprende además:
5 detectar la última imagen entre la pluralidad de imágenes en el segmento de refinamiento progresivo; y
 cesar la codificación según la operación de refinamiento progresivo.
12. El procedimiento de la reivindicación 7, en el que la codificación de al menos algunas de las imágenes en el
10 segmento de refinamiento progresivo según la operación de refinamiento progresivo comprende:
 la aplicación de la operación de refinamiento progresivo a cada imagen en el segmento de refinamiento
 progresivo, a excepción de la última imagen en el segmento de refinamiento progresivo.
13. Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que,
15 cuando son ejecutadas, hacen que un procesador programable de un dispositivo codificador de vídeo realice
 el procedimiento según una cualquiera de las reivindicaciones 1 a 12.
14. Un dispositivo para la decodificación de datos de vídeo, dispositivo que comprende:
20 medios para recibir, en un mensaje de SEI de información de mejora suplementaria, un valor delta del
 recuento del orden de imágenes, POC, que indica una diferencia entre un valor del POC de una imagen
 que precede inmediatamente a una pluralidad de imágenes en un segmento de refinamiento progresivo
 en un flujo de bits codificado, y un valor del POC de una última imagen entre la pluralidad de imágenes en
25 el segmento de refinamiento progresivo; y
 medios para la decodificación, en base al valor recibido de delta de POC, de al menos algunas de las
 imágenes en el segmento de refinamiento progresivo, según una operación de refinamiento progresivo,
 en el que la operación de refinamiento progresivo indica un refinamiento con respecto a la calidad de una
 imagen.

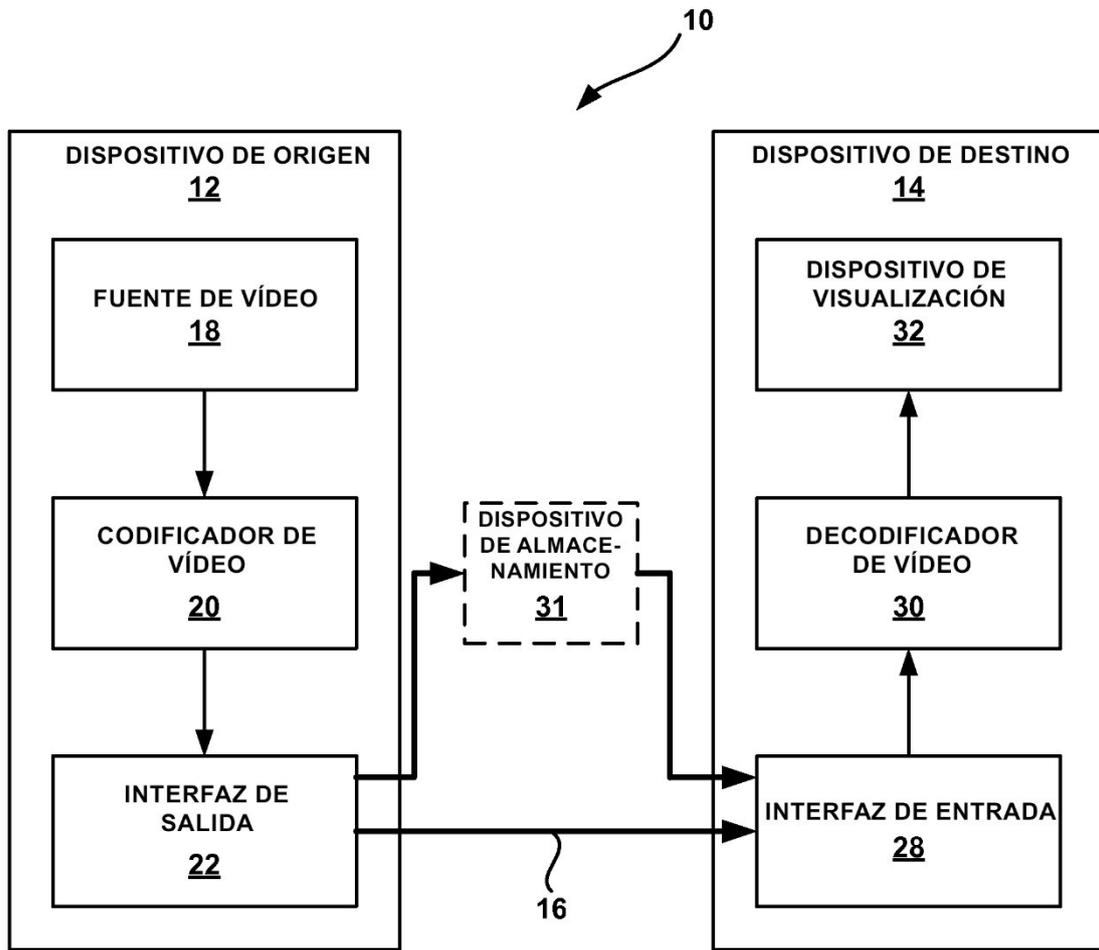


FIG. 1

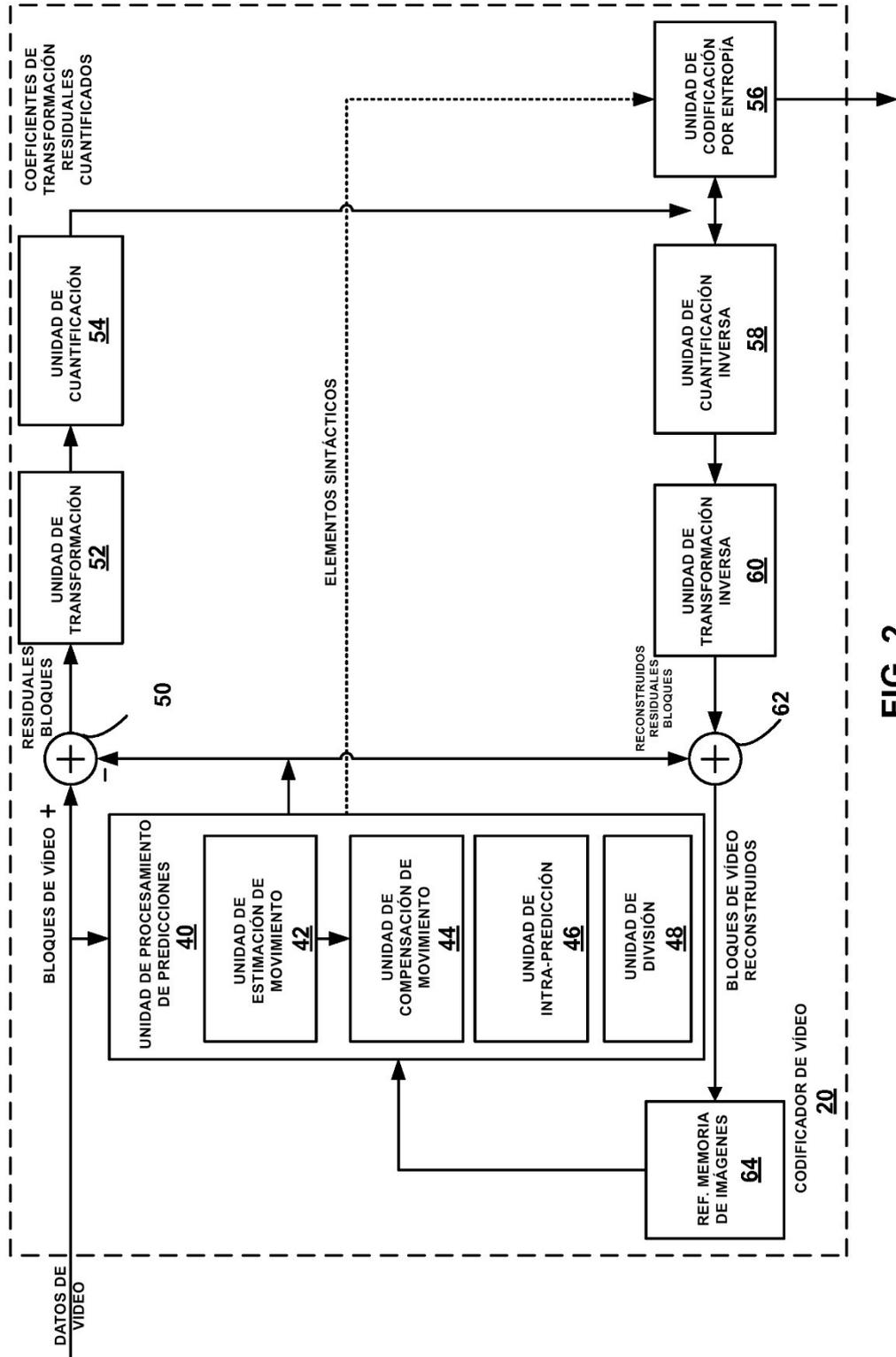


FIG. 2

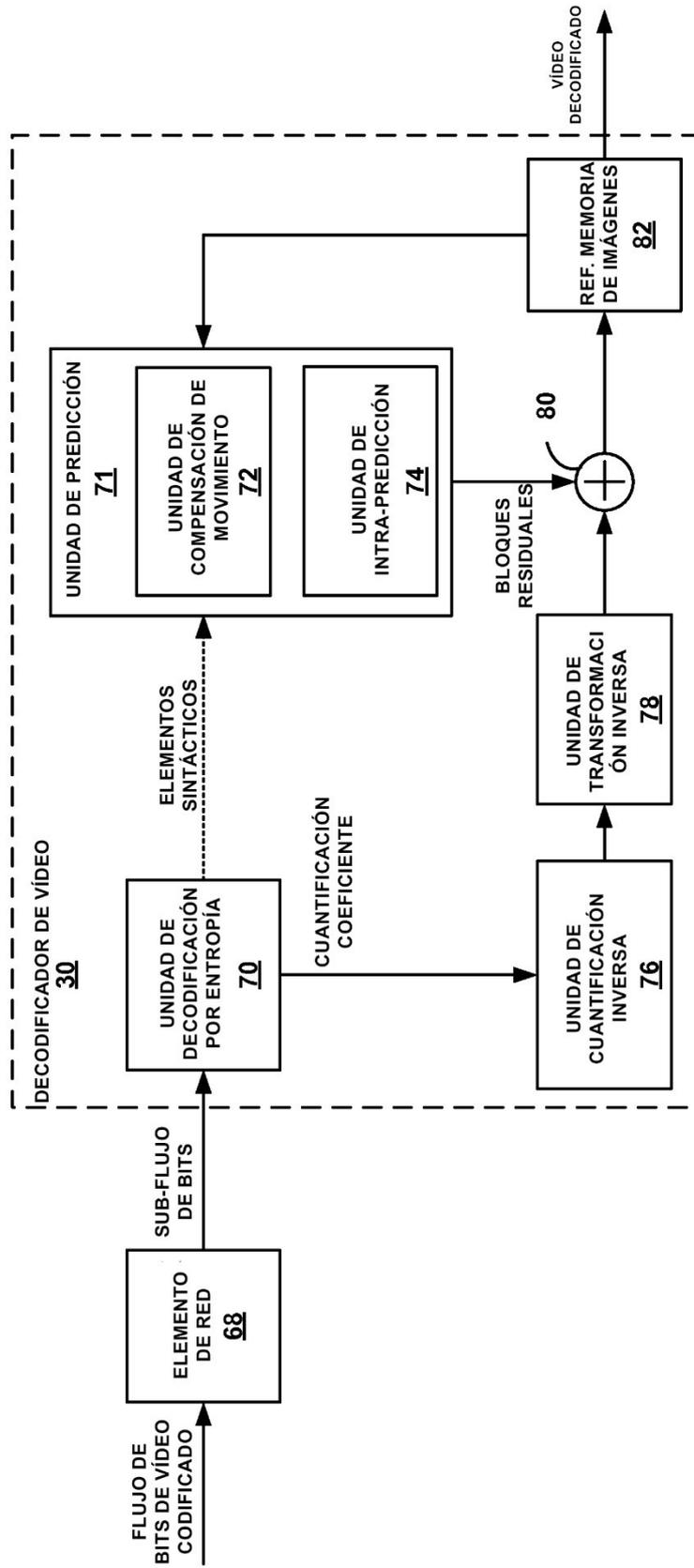


FIG. 3

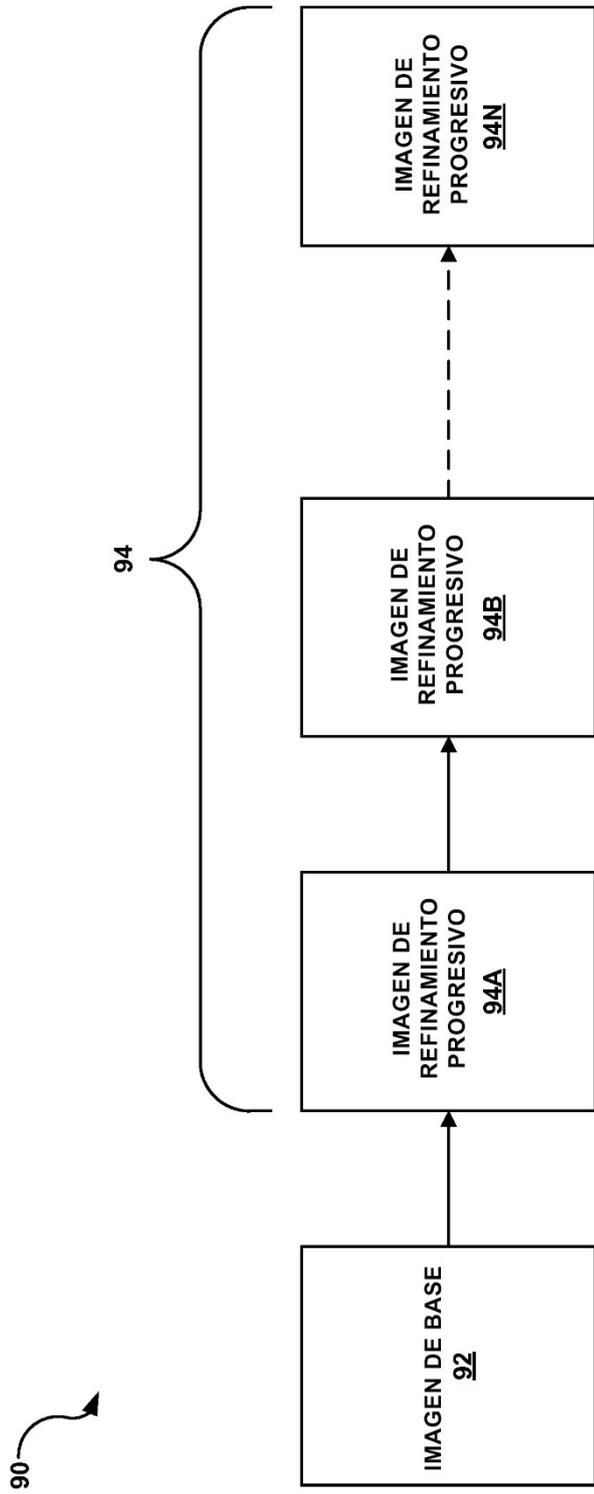


FIG. 4

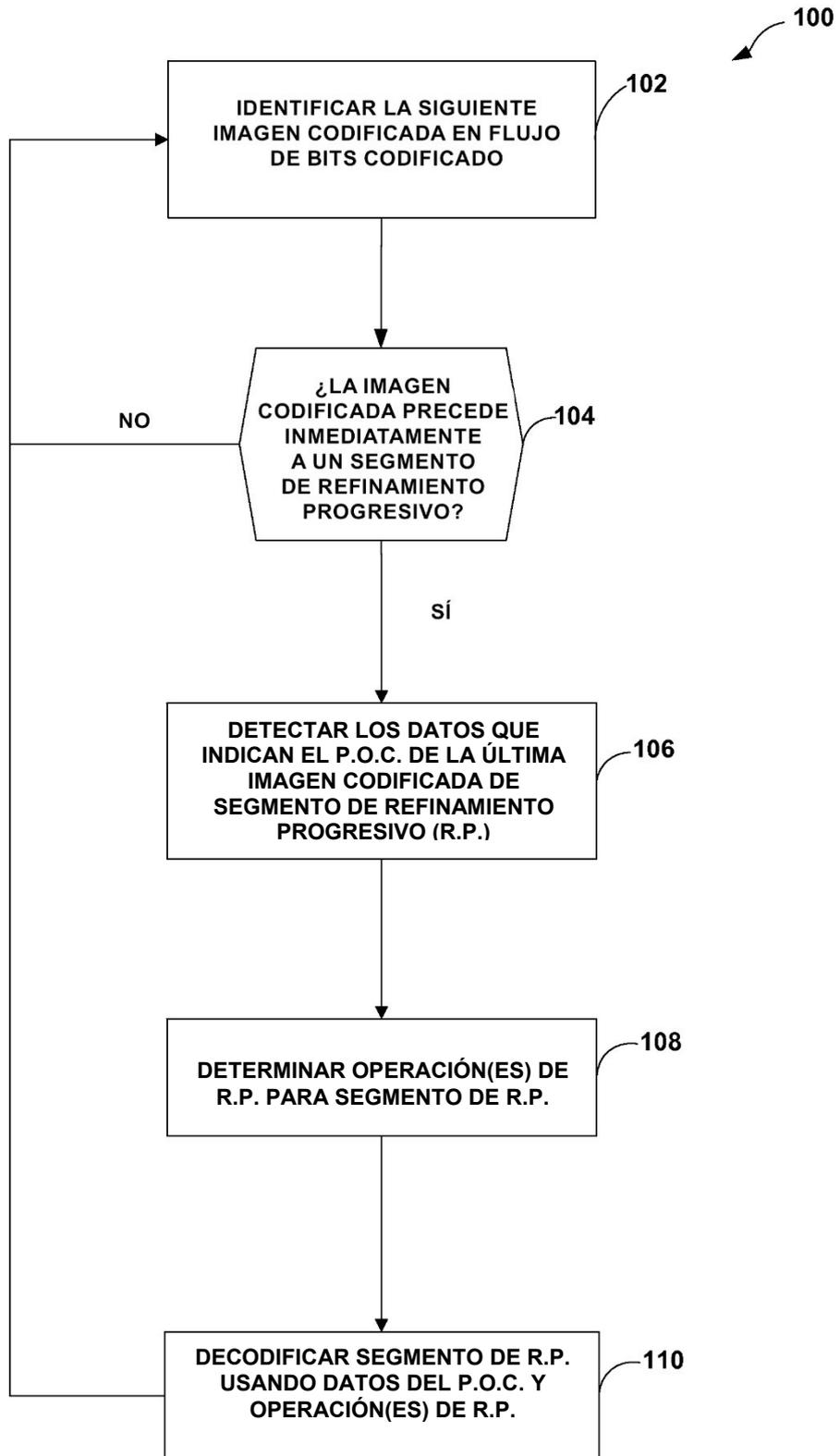


FIG. 5

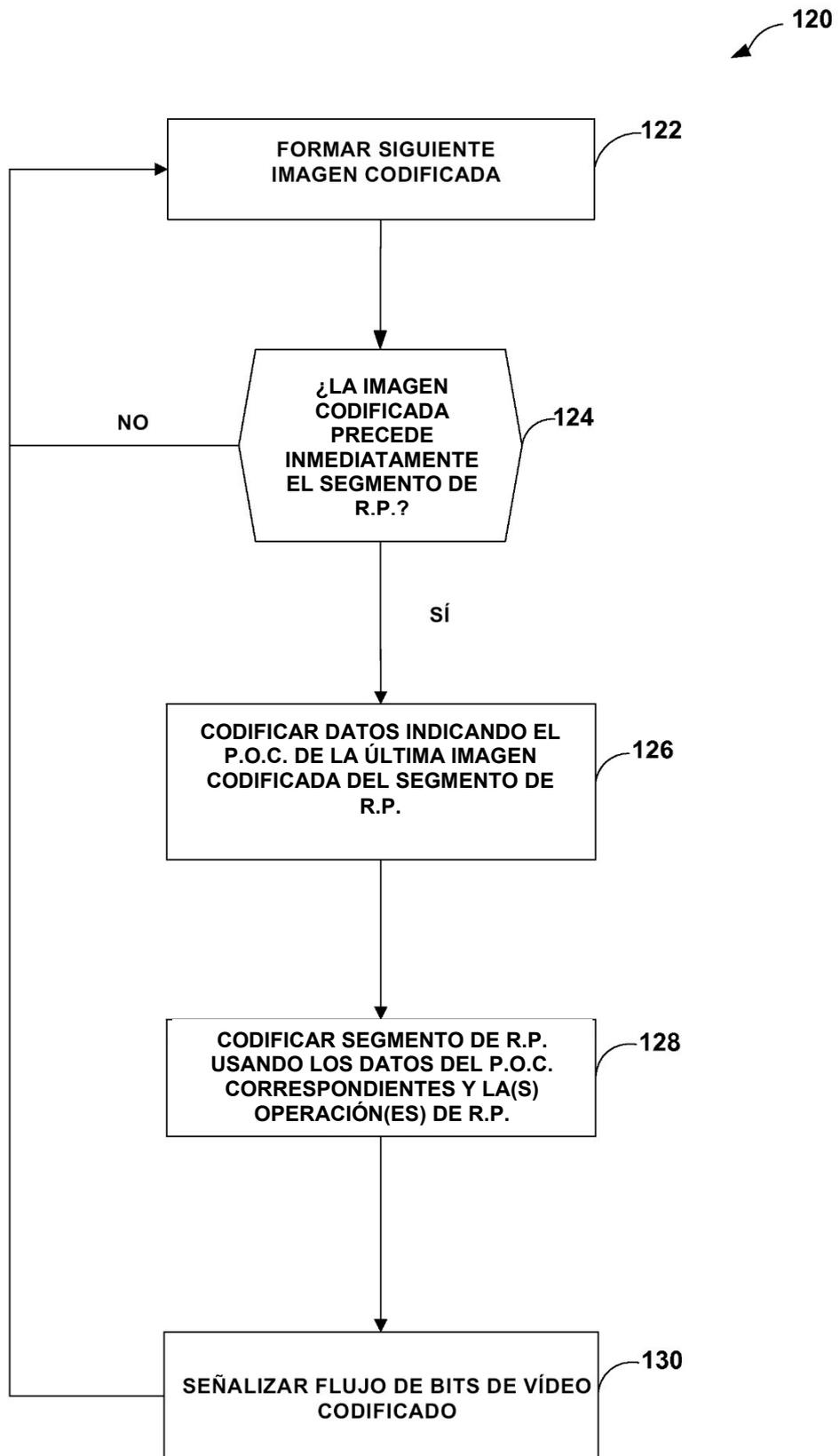


FIG. 6