

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 022 203**

51 Int. Cl.:

H04N 19/52 (2014.01)

H04N 19/70 (2014.01)

H04N 19/102 (2014.01)

H04N 19/136 (2014.01)

H04N 19/157 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **04.10.2017** **PCT/US2017/055091**

87 Fecha y número de publicación internacional: **12.04.2018** **WO18067672**

96 Fecha de presentación y número de la solicitud europea: **04.10.2017** **E 17784520 (3)**

97 Fecha y número de publicación de la concesión europea: **02.04.2025** **EP 3523971**

54 Título: **Precisión del vector de movimiento adaptativo para codificación de vídeo**

30 Prioridad:

04.10.2016 US 201662404105 P

03.10.2017 US 201715724044

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

28.05.2025

73 Titular/es:

QUALCOMM INCORPORATED (100.00%)
International IP Administration 5775 Morehouse
Drive
San Diego, CA 92121-1714, US

72 Inventor/es:

CHEN, JIANLE;
CHIEN, WEI-JUNG;
LI, XIANG;
ZHANG, LI y
KARCZEWICZ, MARTA

74 Agente/Representante:

ISERN JARA, Jorge

ES 3 022 203 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

Precisión del vector de movimiento adaptativo para codificación de vídeo

5 CAMPO TÉCNICO

La presente descripción se refiere a la codificación de vídeo y, más particularmente, a la codificación de vídeo de interpredicción.

10 ANTECEDENTES

Las capacidades de vídeo digital se pueden incorporar en una amplia gama de dispositivos, incluidos los televisores digitales, los sistemas de difusión directa digital, los sistemas de difusión inalámbrica, los asistentes digitales personales (*Personal Digital Assistants*, PDA), los ordenadores portátiles o de escritorio, las cámaras digitales, los dispositivos de grabación digital, los reproductores de medios digitales, los dispositivos de videojuegos, las consolas de videojuegos, los teléfonos celulares o de radio por satélite, los dispositivos de videoconferencia y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263 o ITU-T H.264/MPEG-4, Parte 10, Codificación de vídeo avanzada (*Advanced Video Coding*, AVC), y extensiones de dichos estándares, para transmitir y recibir información de vídeo digital de manera más eficiente.

Las técnicas de compresión de vídeo realizan predicción espacial y/o predicción temporal para reducir o eliminar la redundancia inherente en las secuencias de vídeo. Para codificación de vídeo basada en bloques, un fotograma o fragmento de vídeo puede dividirse en macrobloques. Cada macrobloque se puede dividir aún más. Los macrobloques en un fotograma o fragmento intracodificado (I) se codifican usando predicción espacial con respecto a los macrobloques vecinos. Los macrobloques en un fotograma o fragmento intercodificado (P o B) pueden usar predicción espacial con respecto a los macrobloques vecinos en el mismo fotograma o fragmento o predicción temporal con respecto a otros fotogramas de referencia.

El documento EP 3 203 743 corresponde a una traducción al idioma inglés del documento WO 2016/068674 y describe la determinación de un predictor de vector de movimiento óptimo y la resolución de un vector de movimiento con el fin de codificar o decodificar vídeo de manera eficiente. El documento US2015/0195562 describe técnicas para seleccionar de forma adaptativa la precisión del vector de movimiento para el movimiento de los vectores de movimiento usados para codificar bloques de datos de vídeo y para determinar mediante un decodificador de vídeo la misma precisión del vector de movimiento que el codificador de vídeo seleccionado para cada uno de los bloques.

RESUMEN

La invención se define mediante las reivindicaciones independientes adjuntas. Las características preferidas se definen en las reivindicaciones dependientes.

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

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 usar las técnicas de esta descripción.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar las técnicas de esta descripción.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar las técnicas de esta descripción.

LA FIG. 4 es un diagrama conceptual que ilustra posiciones de píxel fraccionarias para una posición de píxel completa.

Las FIGS. 5A-5C son diagramas conceptuales que ilustran las posiciones de píxeles de crominancia y luminancia correspondientes.

LA FIG. 6 es una ilustración de un ejemplo de coincidencia de plantillas en forma de L para la derivación del vector de movimiento del lado del decodificador (*Decoder side Motion Vector Derivation*, DMVD).

La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de derivación de MV bidireccional basada en espejo.

La FIG. 8 es un diagrama conceptual que ilustra posiciones de píxeles enteros y palabras de código para diferentes precisiones de diferencia de vectores de movimiento.

La FIG. 9 es un diagrama de flujo que ilustra un procedimiento de codificación ejemplar de la descripción.

La FIG. 10 es un diagrama de flujo que ilustra un procedimiento de decodificación ejemplar de la descripción.

DESCRIPCIÓN DETALLADA

Esta descripción se refiere a técnicas para la codificación y señalización de vectores de movimiento (*Motion Vector*, MV) y/o diferencias de vectores de movimiento (*Motion Vector Difference*, MVD), incluidas las técnicas para determinar de forma adaptativa la precisión de MV/MVD. Las técnicas descritas en esta invención pueden usarse en el contexto de códecs de vídeo avanzados, como extensiones de HEVC o la próxima generación de estándares de codificación de vídeo.

Los codificadores de vídeo y decodificadores de vídeo H.264 y H.265 admiten vectores de movimiento con una precisión de un cuarto de píxel. Con una precisión de un cuarto de píxel, la diferencia entre los valores de palabra de código sucesivos para un MV o un MVD es igual a un cuarto de la distancia entre píxeles. Del mismo modo, en la predicción de un octavo de píxel, la diferencia entre los valores de palabra de código sucesivos para un MV o un MVD es igual a un octavo de la distancia entre píxeles. En la precisión de píxeles enteros, la diferencia entre los valores de palabras de código sucesivas para un MV o un MVD es igual a un píxel completo en distancia.

La precisión de píxel usada generalmente no se señala ni se deriva, sino que es fija (por ejemplo, preconfigurada o predeterminada). En algunos casos, la precisión de un octavo de píxel puede proporcionar ciertas ventajas sobre la precisión de un cuarto de píxel o la precisión de píxeles enteros. Sin embargo, la codificación de cada vector de movimiento con una precisión de un octavo de píxel puede usar una gran cantidad de bits de codificación, lo que puede superar los beneficios que de otro modo proporcionarían los vectores de movimiento con una precisión de un octavo de píxel. Es decir, la velocidad de bits requerida para señalar un MV o MVD con una precisión de un octavo de píxel puede superar cualquier ganancia de distorsión. Para algunos tipos de contenido de vídeo, puede ser preferible codificar vectores de movimiento sin interpolación en absoluto, en otras palabras, usando solo precisión de píxeles enteros.

El contenido de la pantalla, como el contenido generado por un ordenador (por ejemplo, texto o gráficos simples), generalmente implica una serie de píxeles que tienen exactamente los mismos valores de píxeles, seguidos de un cambio brusco en los valores de píxeles. Por ejemplo, en el contenido de la pantalla que incluye texto azul sobre un fondo blanco, los píxeles que forman una letra azul pueden tener los mismos valores de píxeles, mientras que el fondo blanco también tiene los mismos valores de píxeles, pero los valores de píxeles blancos pueden ser significativamente diferentes de los valores de píxeles azules. El contenido adquirido por la cámara (por ejemplo, las llamadas imágenes naturales), por el contrario, generalmente incluye cambios lentos en los valores de píxeles debido al movimiento, las sombras, los cambios de iluminación y otros fenómenos naturales. Como el contenido de la pantalla y el contenido adquirido por la cámara generalmente tienen características diferentes, las herramientas de codificación efectivas para un tipo de contenido pueden no ser necesariamente efectivas para el otro tipo de contenido. Como un ejemplo, la interpolación de subpíxeles para la codificación de interpredicción puede mejorar la codificación del contenido de la cámara, pero la complejidad asociada y la sobrecarga de señalización pueden reducir realmente la calidad de codificación y/o la eficiencia del ancho de banda para el contenido de la pantalla.

En algunos ejemplos, las técnicas de esta descripción pueden incluir determinar de forma adaptativa la precisión del vector de movimiento según, por ejemplo, el contenido (por ejemplo, el tipo de contenido) del vídeo que se codifica. En algunos ejemplos, las técnicas de esta descripción pueden incluir derivar, mediante un codificador, una precisión de vector de movimiento apropiada para el contenido de vídeo que se está codificando. Usando las mismas técnicas de derivación, un decodificador de vídeo también puede determinar, sin recibir un elemento sintáctico que indique la precisión del vector de movimiento, qué precisión del vector de movimiento se usó para codificar los datos de vídeo. En otros ejemplos, un codificador de vídeo puede señalar (y el decodificador de vídeo puede recibir), en el flujo de bits de vídeo codificado, la precisión del vector de movimiento seleccionada por el codificador de vídeo.

La selección adaptativa de la precisión del vector de movimiento puede mejorar la calidad general de la codificación de vídeo al permitir que se utilicen vectores de movimiento de mayor precisión (por ejemplo, vectores de movimiento de 1/4 o 1/8 de precisión) para el contenido de vídeo, donde el uso de dicho vector de movimiento de mayor precisión mejora la calidad de codificación de vídeo, por ejemplo, al producir una mejor compensación entre velocidad y distorsión. La selección adaptativa de la precisión del vector de movimiento también puede mejorar la calidad general de la codificación de vídeo al permitir el uso de vectores de movimiento de menor precisión (por ejemplo, precisión de enteros) para el contenido de vídeo donde el uso de vectores de movimiento de mayor precisión no mejora, o incluso empeora, la calidad de la codificación de vídeo.

Varias técnicas en esta descripción se pueden describir con referencia a un codificador de vídeo, que pretende ser un término genérico que puede referirse a un codificador de vídeo o un decodificador de vídeo. A menos que se indique explícitamente lo contrario, no se debe suponer que las técnicas descritas con respecto a un codificador de vídeo o un decodificador de vídeo no pueden ser realizadas por el otro de un codificador de vídeo o un decodificador de vídeo. Por ejemplo, en muchos casos, un decodificador de vídeo realiza la misma técnica de codificación, o a veces una recíproca, que un codificador de vídeo para decodificar datos de vídeo codificados. En muchos casos, un codificador de vídeo también incluye un bucle de decodificación de vídeo y, por lo tanto, el codificador de vídeo realiza la decodificación de vídeo como parte de la codificación de datos de vídeo. Por lo tanto, a menos que se indique lo

contrario, las técnicas descritas en esta descripción con respecto a un decodificador de vídeo también se pueden realizar mediante un codificador de vídeo, y viceversa.

Esta descripción también puede usar términos tales como capa actual, bloque actual, imagen actual, fragmento actual, etc. En el contexto de esta descripción, el término actual pretende identificar una capa, bloque, imagen, fragmento, etc. que se está codificando actualmente, a diferencia de, por ejemplo, capas, bloques, imágenes y fragmentos previamente codificados o bloques, imágenes y fragmentos aún por codificar.

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo 10 que puede usar las técnicas de esta descripción para codificar MV y/o MVD. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo fuente 12 que transmite vídeo codificado a un dispositivo de destino 14 a través de un canal de comunicación 16. El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos. En algunos casos, el dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender dispositivos de comunicación inalámbricos, tales como teléfonos inalámbricos, los llamados radiotelefonos celulares o satelitales, o cualquier dispositivo inalámbrico que pueda comunicar información de vídeo a través de un canal de comunicación 16, en cuyo caso el canal de comunicación 16 es inalámbrico. Sin embargo, las técnicas de esta descripción, que generalmente se refieren a técnicas para admitir la precisión de subpíxeles adaptativos para vectores de movimiento, no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Por ejemplo, estas técnicas pueden aplicarse a difusiones televisivas por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por Internet, vídeo digital codificado que se codifica en un medio de almacenamiento u otros escenarios. Por consiguiente, el canal de comunicación 16 puede comprender cualquier combinación de medios inalámbricos o por cable adecuados para la transmisión de datos de vídeo codificados.

En el ejemplo de la FIG. 1, el dispositivo fuente 12 incluye una fuente de vídeo 18, un codificador de vídeo 20, un modulador/demodulador (módem) 22 y un transmisor 24. El dispositivo de destino 14 incluye un receptor 26, un módem 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. Según esta descripción, el codificador de vídeo 20 del dispositivo fuente 12 puede configurarse para aplicar las técnicas para admitir dos o más técnicas para codificar y/o señalizar MV y/o MVD. En otros ejemplos, un dispositivo fuente y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 12 puede recibir datos de vídeo de una fuente de vídeo externa 18, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema ilustrado 10 de la FIG. 1 es solo un ejemplo. Las técnicas de esta descripción para codificar y/o señalizar MV y/o MVD se pueden realizar mediante cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque generalmente las técnicas de esta descripción se realizan mediante un dispositivo de codificación de vídeo, las técnicas también se pueden realizar mediante un codificador/decodificador de vídeo, normalmente denominado "CODEC". Además, las técnicas de esta descripción también se pueden realizar mediante un preprocesador de vídeo. El dispositivo fuente 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación donde el dispositivo fuente 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de una manera sustancialmente simétrica de manera que cada uno de los dispositivos 12, 14 incluya componentes de codificación y decodificación de vídeo. Por lo tanto, el sistema 10 puede admitir la transmisión de vídeo unidireccional o bidireccional entre el dispositivo fuente 12 y el dispositivo de destino 14, por ejemplo, para transmisión de vídeo, reproducción de vídeo, retransmisión de vídeo o videotelefonía.

La fuente de vídeo 18 del dispositivo fuente 12 puede incluir un dispositivo de captura de vídeo, como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una transmisión de vídeo de un proveedor de contenido de vídeo. Como alternativa adicional, la fuente de vídeo 18 puede generar datos basados en gráficos informáticos como el vídeo de origen, o una combinación de vídeo en vivo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una cámara de vídeo, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o teléfonos de vídeo. Sin embargo, como se mencionó anteriormente, las técnicas descritas en esta descripción se pueden aplicar a la codificación de vídeo en general, y se pueden usar en aplicaciones inalámbricas y/o por cable. En cada caso, el vídeo capturado, precapturado o generado por ordenador puede codificarse mediante el codificador de vídeo 20. A continuación, la información de vídeo codificada puede modularse mediante el módem 22 según una norma de comunicación y transmitirse al dispositivo de destino 14 a través del transmisor 24. El módem 22 puede incluir varios mezcladores, filtros, amplificadores u otros componentes diseñados para la modulación de señal. El transmisor 24 puede incluir circuitos diseñados para transmitir datos, incluidos amplificadores, filtros y una o más antenas.

El receptor 26 del dispositivo de destino 14 recibe información a través del canal 16, y el módem 28 demodula la información. De nuevo, el proceso de codificación de vídeo puede implementar una o más de las técnicas descritas en esta invención para codificar MV y/o MVD. La información comunicada a través del canal 16 puede incluir información de sintaxis definida por el codificador de vídeo 20, que también es usada por el decodificador de vídeo 30, que incluye elementos de sintaxis que describen características y/o procesamiento de macrobloques y otras unidades codificadas, por ejemplo, grupos de imágenes (*Group Of Pictures*, GOP). El dispositivo de visualización 32 muestra

los datos de vídeo decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (*Cathode Ray Tube*, CRT), una pantalla de cristal líquido (*Liquid Cristal Display*, LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (*Organic Light Emitting Diode*, OLED) u otro tipo de dispositivo de visualización.

En el ejemplo de la FIG. 1, el canal de comunicación 16 puede comprender cualquier medio de comunicación inalámbrico o cableado, como un espectro de radiofrecuencia (*Radio Frequency*, RF) o una o más líneas de transmisión físicas, o cualquier combinación de medios inalámbricos y cableados. El canal de comunicación 16 puede formar parte de una red basada en paquetes, como una red de área local (*Local Area Network*, LAN), una red de área extensa (*Wide Area Network*, WAN) o una red global como Internet. El canal de comunicación 16 generalmente representa cualquier medio de comunicación adecuado, o colección de diferentes medios de comunicación, para transmitir datos de vídeo desde el dispositivo fuente 12 al dispositivo de destino 14, incluida cualquier combinación adecuada de medios cableados o inalámbricos. El canal de comunicación 16 puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que facilite la comunicación desde el dispositivo fuente 12 al dispositivo de destino 14.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar según una norma de compresión de vídeo. Sin embargo, las técnicas de la presente descripción no se limitan a ninguna norma de codificación particular. Los estándares de ejemplo incluyen el estándar ITU-T H.265, el estándar ITU-T H.264, también conocido como MPEG-4, Parte 10, Codificación de vídeo avanzada (*Advanced Video Coding*, AVC), MPEG-2 e ITU-T H.263. Aunque no se muestra en la FIG. 1, en algunos ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden integrarse cada uno con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación de flujos de audio y vídeo en un flujo de datos común o en flujos de datos separados. Si corresponde, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor ITU H.223 u otros protocolos como el protocolo de datagramas de usuario (*User Datagram Protocol*, UDP).

Los estándares de codificación de vídeo también incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual y ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluyendo sus extensiones de Codificación de Vídeo Escalable (*Scalable Video Coding*, SVC) y Codificación de Vídeo Multivista (*Multi-view Video Coding*, MVC).

Además, el Equipo de Colaboración Conjunta sobre Codificación de Vídeo (*Joint Collaboration Team on Video Coding*, JCTVC) del Grupo de Expertos en Codificación de Vídeo (*Video Coding Experts Group*, VCEG) de ITU-T y el Grupo de Expertos en Cine (*Motion Picture Experts Group*, MPEG) de ISO/IEC ha desarrollado un nuevo estándar de codificación de vídeo, denominado Codificación de Vídeo de Alta Eficiencia (*High Efficiency Video Coding*, HEVC). El último borrador de especificación HEVC, al que en adelante se hará referencia como HEVC WD, está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/14Vienna/wg11/JCTVC-N1003-v1.zip. Una copia de la norma HEVC finalizada (es decir, ITU-T H.265, Serie H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services - Coding of moving video, abril de 2015) está disponible en <https://www.itu.int/rec/T-REC-H.265-201504-I/en>.

ITU-T VCEG (Q6/16) e ISO/IEC MPEG (JTC 1/SC 29/WG11) están estudiando ahora la necesidad potencial de estandarización de la futura tecnología de codificación de vídeo con una capacidad de compresión que supera significativamente la del estándar HEVC actual (incluidas sus extensiones actuales y las extensiones a corto plazo para la codificación de contenido de pantalla y la codificación de alto rango dinámico). Los grupos están trabajando juntos en esta actividad de exploración en un esfuerzo conjunto de colaboración conocido como el Equipo Conjunto de Exploración de Vídeo (*Joint Video Exploration Team*, JVET) para evaluar los diseños de tecnología de compresión propuestos por sus expertos en esta área. El JVET se reunió por primera vez del 19 al 21 de octubre de 2015. La última versión del software de referencia, es decir, el Modelo de Exploración Conjunta 3 (JEM 3.0), se podía descargar desde: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-3.0/

Una descripción del algoritmo del Modelo de Prueba de Exploración Conjunta 3 (JEM3) se describe en J. Chen, y col., "Algorithm Description of Joint Exploration Test Model 3," Joint Video Exploration Team (*Joint Video Exploration Team*, JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 3rd Meeting: Ginebra, CH, 26 de mayo a 1 de junio de 2016, JVET-C1001.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse como circuitos codificadores o decodificadores adecuados, como uno o más microprocesadores, procesadores de señales digitales (*Digital Signal Processors* DSP), circuitos integrados de aplicación específica (*Application Specific Integrated Circuits* ASIC), matrices de puertas programables en campo (*Field Programmable Gate Arrays*, FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de estos. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar

incluidos en uno o más codificadores o decodificadores, cada uno de los cuales puede integrarse como parte de un codificador/decodificador combinado (*Combined Encoder/Decoder*, CODEC) en una cámara, ordenador, dispositivo móvil, dispositivo de suscriptor, dispositivo de transmisión, decodificador, servidor o similar.

Una secuencia de vídeo suele incluir una serie de fotogramas de vídeo. Un grupo de imágenes (*Group Of Pictures*, GOP) generalmente comprende una serie de uno o más fotogramas de vídeo. Un GOP puede incluir datos de sintaxis en un encabezado del GOP, en el encabezado de uno o más fotogramas del GOP, o en cualquier otro lugar, que describan el número de fotogramas incluidos en el GOP. Cada fotograma puede incluir datos de sintaxis que describen un modo de codificación para el fotograma respectivo. El codificador de vídeo 20 suele operar con bloques de vídeo dentro de fotogramas individuales para codificar los datos de vídeo. Para H.264, un bloque de vídeo puede corresponder a un macrobloque o una partición de un macrobloque. Los bloques de vídeo pueden tener tamaños fijos o variables, y pueden diferir en tamaño según una norma de codificación especificada. Cada fotograma de vídeo puede incluir una pluralidad de fragmentos. Cada fragmento puede incluir una pluralidad de macrobloques, que pueden estar dispuestos en particiones, también denominados subbloques.

Como ejemplo, el estándar ITU-T H.264 admite la intrapredicción en varios tamaños de bloque, como 16 por 16, 8 por 8 o 4 por 4 para componentes de luminancia y 8x8 para componentes de crominancia, así como la interpredicción en varios tamaños de bloque, como 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 y 4x4 para componentes de luminancia y tamaños escalados correspondientes para componentes de crominancia. En esta descripción, "NxN" y "N por N" pueden usarse indistintamente para referirse a las dimensiones en píxeles del bloque, tanto verticales como horizontales; por ejemplo, 16x16 píxeles o 16x16 píxeles. En general, un bloque de 16x16 tendrá 16 píxeles en dirección vertical ($y=16$) y 16 píxeles en dirección horizontal ($x=16$). Del mismo modo, un bloque NxN generalmente tiene N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque pueden disponerse en filas y columnas. Además, los bloques no necesitan tener necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden estar compuestos de NxM píxeles, donde M no es necesariamente igual a N.

Los tamaños de bloque que son menores que 16 por 16 pueden denominarse particiones de un macrobloque de 16 por 16. Los bloques de vídeo pueden comprender bloques de datos de píxeles en el dominio de píxeles, o bloques de coeficientes de transformación en el dominio de transformación, por ejemplo, después de la aplicación de una transformación tal como una transformación de coseno discreta (*Discrete Cosine Transform*, DCT), una transformación entera, una transformación de ondícula o una transformación conceptualmente similar a los datos de bloques de vídeo residuales que representan diferencias de píxeles entre bloques de vídeo codificados y bloques de vídeo predictivos. En algunos casos, un bloque de vídeo puede comprender bloques de coeficientes de transformada cuantificados en el dominio de transformada.

Los bloques de vídeo más pequeños pueden proporcionar una mejor resolución y pueden usarse para ubicaciones de un fotograma de vídeo que incluyen altos niveles de detalle. En general, los macrobloques y las diversas particiones, a veces denominados subbloques, pueden considerarse bloques de vídeo. Además, un fragmento puede considerarse como una pluralidad de bloques de vídeo, tales como macrobloques y/o subbloques. Cada fragmento puede ser una unidad decodificable independientemente de un fotograma de vídeo. Alternativamente, los fotogramas en sí mismos pueden ser unidades decodificables, u otras partes de un fotograma pueden definirse como unidades decodificables. El término "unidad codificada" puede referirse a cualquier unidad decodificable independientemente de un fotograma de vídeo, como un fotograma completa, un fragmento de un fotograma, un grupo de imágenes (GOP), también denominada secuencia, u otra unidad decodificable independientemente definida según las técnicas de codificación aplicables.

Como se analizó anteriormente, se ha finalizado una nueva norma de codificación de vídeo, denominada ITU-T H.265, Codificación de vídeo de alta eficiencia (*High Efficiency Video Coding*, HEVC). Las extensiones a HEVC incluyen la extensión de codificación de contenido de pantalla. Los esfuerzos de estandarización de HEVC se basaron en un modelo de dispositivo de codificación de vídeo, denominado Modelo de Prueba HEVC (HM). El HM presupone diversas capacidades de los dispositivos de codificación de vídeo según, por ejemplo, la norma ITU-T H.264/AVC. Por ejemplo, mientras que H.264 ofrece nueve modos de codificación de intrapredicción, HM ofrece hasta treinta y tres.

HM se refiere a un bloque de datos de vídeo como una unidad de codificación (*Coding Unit*, CU). Los datos de sintaxis dentro de un flujo de bits pueden definir una unidad de codificación máxima (*Largest Coding Unit*, LCU), que es la unidad de codificación máxima en términos de número de píxeles. En general, una UC tiene una función similar a la de un macrobloque de H.264, salvo que no tiene distinción de tamaño. Por lo tanto, un CU puede dividirse en sub-CU. En general, las referencias en esta descripción a una CU pueden referirse a una unidad de codificación más grande de una imagen o una sub-CU de una LCU. Una LCU se puede dividir en sub-CU, y cada sub-CU se puede dividir en sub-CU. Los datos sintácticos para un flujo de bits pueden definir un número máximo de veces que se puede dividir una LCU, denominado profundidad de CU. Por consiguiente, un flujo de bits también puede definir una unidad de codificación más pequeña (*Smallest Coding Unit*, SCU). Esta descripción también puede usar el término "bloque" para

referirse a cualquiera de una CU, PU o TU. Además, cuando esta descripción se refiere a ejemplos que implican una unidad de codificación o CU, debe entenderse que pueden proporcionarse otros ejemplos con respecto a macrobloques sustituidos por unidades de codificación.

Una LCU puede estar asociada con una estructura de datos de árbol cuádruple. En general, una estructura de datos de árbol cuaternario incluye un nodo por CU, donde un nodo raíz corresponde a la LCU. Si una CU se divide en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU. Cada nodo de la estructura de datos de árbol cuádruple puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo en el árbol cuádruple puede incluir un indicador de división, que indica si la CU correspondiente al nodo se divide en sub-CU. Los elementos de sintaxis para una CU pueden definirse de forma recursiva y pueden depender de si la CU se divide en sub-CU.

Un CU que no está dividido (por ejemplo, correspondiente a un nodo hoja en la estructura de datos de árbol cuádruple) puede incluir una o más unidades de predicción (*Prediction Units*, PU). En general, una PU representa la totalidad o una parte de la CU correspondiente e incluye datos para recuperar una muestra de referencia para la PU. Por ejemplo, cuando la PU tiene codificación intramodo, puede incluir datos que describen un modo de intrapredicción para la PU. Como otro ejemplo, cuando la PU tiene codificación intermodo, puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento 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 (p. ej., precisión de píxel entero, precisión de un cuarto de píxel, precisión de un octavo de píxel), un fotograma de referencia al que apunta el vector de movimiento y/o una lista de referencia (p. ej., lista 0 o lista 1) para el vector de movimiento. Los datos para la CU que definen las PU también pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de partición pueden variar según si la UC no está codificada, está codificada en modo intrapredicción o está codificada en modo interpredicción.

Una UC con una o más PU también puede incluir una o más unidades de transformación (*Transform Units*, TU). Después de la predicción usando una PU, un codificador de vídeo puede calcular un valor residual para la parte de la CU correspondiente a la PU. El valor residual se puede transformar, cuantificar y escanear. Una TU no se limita necesariamente al tamaño de una PU. Por lo tanto, las TU pueden ser mayores o menores que las PU correspondientes para la misma CU. En algunos ejemplos, el tamaño máximo de una TU puede corresponder al tamaño de la CU que incluye la TU.

Codificar una PU usando interpredicción implica calcular un vector de movimiento entre un bloque actual y un bloque en un fotograma de referencia. Los vectores de movimiento se calculan a través de un proceso llamado estimación de movimiento (o búsqueda de movimiento). Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una unidad de predicción en un fotograma actual con respecto a una muestra de referencia de un fotograma de referencia. Una muestra de referencia puede ser un bloque que se asemeja estrechamente a la parte de la UC, incluida la PU, que se codifica en términos de diferencia de píxeles, la cual puede determinarse mediante la suma de la diferencia absoluta (*Sum of Absolute Difference*, SAD), la suma de la diferencia al cuadrado (*Sum of Squared Difference*, SSD) u otras métricas de diferencia. La muestra de referencia puede ocurrir en cualquier lugar dentro de un fotograma de referencia o fragmento de referencia. En algunos ejemplos, la muestra de referencia puede ocurrir en una posición de píxel fraccional. Al encontrar una parte del fotograma de referencia que mejor coincide con la parte actual, el codificador determina el vector de movimiento actual para la parte actual como la diferencia en la ubicación de la parte actual a la parte coincidente en el fotograma de referencia (es decir, desde el centro de la parte actual al centro de la parte coincidente).

En algunos ejemplos, el codificador de vídeo 20 puede señalar el vector de movimiento para cada parte en el flujo de bits de vídeo codificado. El vector de movimiento señalado es usado por el decodificador de vídeo 30 para realizar la compensación de movimiento con el fin de decodificar los datos de vídeo. Sin embargo, la señalización del vector de movimiento original directamente puede dar como resultado una codificación menos eficiente, ya que normalmente se necesita un gran número de bits para transmitir la información.

En algunos ejemplos, en lugar de señalar directamente el vector de movimiento original, el codificador de vídeo 20 puede predecir un vector de movimiento para cada partición o bloque de vídeo (por ejemplo, para cada PU en HEVC). Al realizar esta predicción de vector de movimiento, el codificador de vídeo 20 puede seleccionar un conjunto de vectores de movimiento candidatos determinados a partir de bloques espacialmente vecinos en el mismo fotograma que la parte actual o un vector de movimiento candidato determinado a partir de un bloque cúbicado en un fotograma de referencia (por ejemplo, un predictor de vector de movimiento temporal (MVP)). El codificador de vídeo 20 puede realizar una predicción de vectores de movimiento y, si es necesario, señalar la diferencia de predicción (también llamada diferencia de vectores de movimiento (MVD)), en lugar de señalar un vector de movimiento original, para reducir la tasa de bits en la señalización. Los vectores de vector de movimiento candidatos de los bloques espacialmente vecinos pueden denominarse candidatos MVP espaciales, mientras que el vector de movimiento

candidato del bloque cublicado en otro fotograma de referencia puede denominarse candidato MVP temporal.

En un sistema de codificación de vídeo basado en bloques, tal como las normas de codificación de vídeo HEVC y H.264/AVC, para cada bloque, puede estar disponible un conjunto de información de movimiento, y se usa para generar los resultados de interpredicción usando un proceso de compensación de movimiento. Por lo general, un conjunto de información de movimiento contiene información de movimiento para direcciones de predicción hacia adelante y hacia atrás. Aquí, las direcciones de predicción hacia adelante y hacia atrás son dos direcciones de predicción de un modo de predicción bidireccional y los términos "hacia adelante" y "hacia atrás" no necesariamente tienen un significado de geometría; en cambio, corresponden a la lista de imágenes de referencia 0 (RefPicList0) y la lista de imágenes de referencia 1 (RefPicList1) de una imagen actual. Cuando solo está disponible una lista de imágenes de referencia para una imagen o fragmento, solo está disponible RefPicList0 y la información de movimiento de cada bloque de un fragmento siempre está hacia adelante.

Para cada dirección de predicción, la información de movimiento generalmente contiene un índice de referencia y un vector de movimiento. En algunos casos, por simplicidad, se puede hacer referencia a un vector de movimiento en sí mismo de manera que se suponga que tiene un índice de referencia asociado. Se usa un índice de referencia para identificar una imagen de referencia en la lista de imágenes de referencia actual (RefPicList0 o RefPicList1). Un vector de movimiento tiene un componente horizontal y uno vertical.

En el estándar HEVC, hay dos modos de interpredicción, denominados modos de fusión (skip se considera un caso especial de fusión) y predicción avanzada de vectores de movimiento (AMVP), respectivamente, para una unidad de predicción (PU).

En el modo AMVP o de fusión, se mantiene una lista de candidatos a predictor de vector de movimiento (MV) para múltiples predictores de vector de movimiento. Los vectores de movimiento, así como los índices de referencia en el modo de fusión, de la PU actual se generan tomando un candidato de la lista de candidatos de MV. En el modo de fusión, un candidato de fusión se identifica mediante un índice de fusión, las imágenes de referencia se usan para la predicción de los bloques actuales, así como los vectores de movimiento asociados, y no se señala más información de diferencia de vector de movimiento (MVD). En el modo AMVP, para cada dirección de predicción potencial de la lista 0 o la lista 1, se señala explícitamente un índice de referencia, junto con un índice MVP a la lista de candidatos MV, y la información MVD se señala adicionalmente para refinar la información MV del PU actual.

En el modo de fusión, el codificador de vídeo 20 indica al decodificador de vídeo 30, a través de la señalización de flujo de bits de la sintaxis de predicción, que copie un vector de movimiento, un índice de referencia (que identifica un fotograma de referencia, en una lista de imágenes de referencia dada, a la que apunta el vector de movimiento) y la dirección de predicción de movimiento (que identifica la lista de imágenes de referencia (Lista 0 o Lista 1), es decir, en términos de si el fotograma de referencia precede o sigue temporalmente al fotograma codificado actualmente) de un vector de movimiento candidato seleccionado para una parte actual del fotograma. Esto se logra mediante la señalización en el flujo de bits de un índice en una lista de vectores de movimiento candidatos que identifica el vector de movimiento candidato seleccionado (es decir, el candidato de MVP espacial o candidato de MVP temporal particular). Por lo tanto, para el modo de fusión, la sintaxis de predicción puede incluir un indicador que identifique el modo (en este caso, el modo de "fusión") y un índice que identifique el vector de movimiento candidato seleccionado. En algunos casos, el vector de movimiento candidato estará en una parte causal en referencia a la parte actual. Es decir, el vector de movimiento candidato ya habrá sido decodificado por el decodificador de vídeo 30. Como tal, el decodificador de vídeo 30 ya ha recibido y/o determinado el vector de movimiento, el índice de referencia y la dirección de predicción de movimiento para la parte causal. Como tal, el decodificador puede simplemente recuperar el vector de movimiento, el índice de referencia y la dirección de predicción de movimiento asociados con la parte causal de la memoria y copiar estos valores como la información de movimiento para la parte actual. Para reconstruir un bloque en modo de fusión, el decodificador obtiene el bloque predictivo usando la información de movimiento derivada para la parte actual, y añade los datos residuales al bloque predictivo para reconstruir el bloque codificado.

En AMVP, el codificador de vídeo 20 instruye al decodificador de vídeo 30, a través de la señalización de flujo de bits, para copiar solo el vector de movimiento de la parte candidata y usar el vector copiado como un predictor para el vector de movimiento de la parte actual. El codificador de vídeo 20 también codifica y señala un MVD al decodificador de vídeo 30 junto con el índice de referencia del fotograma de referencia para el predictor de vector de movimiento de la dirección de predicción asociada con el predictor de vector de movimiento del bloque actual. Un MVD es la diferencia entre el vector de movimiento actual para el bloque actual y un predictor de vector de movimiento derivado de un bloque candidato. En este caso, el codificador de vídeo 20, usando la estimación de movimiento, determina un vector de movimiento real para el bloque que se va a codificar, y luego determina la diferencia entre el vector de movimiento real y el predictor de vector de movimiento como el valor MVD. De esta manera, el decodificador de vídeo 30 no usa una copia exacta del vector de movimiento candidato como el vector de movimiento actual, como en el modo de fusión, sino que puede usar un vector de movimiento candidato que puede tener un valor "cercano" al vector de movimiento actual determinado a partir de la estimación de movimiento y añadir el MVD para reproducir el vector de movimiento actual. Para reconstruir un bloque en modo AMVP, el decodificador de vídeo 30 añade los datos residuales correspondientes al bloque señalado por el vector de movimiento actual para reconstruir el bloque codificado.

En la mayoría de las circunstancias, el MVD requiere menos bits para señalar que todo el vector de movimiento actual. Como tal, AMVP permite una señalización más precisa del vector de movimiento actual mientras mantiene la eficiencia de codificación sobre el envío de todo el vector de movimiento. Por el contrario, el modo de fusión no permite la especificación de un MVD y, como tal, el modo de fusión sacrifica la precisión de la señalización del vector de movimiento para una mayor eficiencia de señalización (es decir, menos bits). La sintaxis de predicción para AMVP puede incluir un indicador para el modo (en este caso, el indicador de AMVP), el índice para el bloque candidato, el MVD entre el vector de movimiento actual y el vector de movimiento predictivo del bloque candidato, el índice de referencia y la dirección de predicción de movimiento.

Según las técnicas de ejemplo de esta descripción, el codificador de vídeo 20 puede codificar intermodo una CU (por ejemplo, usando interpredicción) usando una o más PU que tienen vectores de movimiento de precisión de píxel subentero y/o entero variable. Por ejemplo, el codificador de vídeo 20 puede seleccionar entre usar un vector de movimiento que tenga una precisión de píxel entero o una precisión de píxel fraccional (por ejemplo, un cuarto o un octavo) para una PU según el contenido de los datos de vídeo que se codifican. Según algunas técnicas de esta descripción, el codificador de vídeo 20 puede no necesitar generar, para su inclusión en el flujo de bits de datos de vídeo codificados, una indicación de la precisión de subpíxel para un vector de movimiento de una PU. En cambio, el decodificador de vídeo 30 puede derivar la precisión del vector de movimiento usando las mismas técnicas de derivación usadas por el codificador de vídeo 20. Según otras técnicas de esta descripción, el codificador de vídeo 20 puede incluir, en el flujo de bits de datos de vídeo codificados, uno o más elementos sintácticos que el decodificador de vídeo 30 puede usar para determinar la precisión del vector de movimiento seleccionado.

Según las técnicas de la descripción descritas con más detalle a continuación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para determinar una diferencia de vector de movimiento (MVD) para un bloque actual de datos de vídeo usando una de una pluralidad de precisiones de MVD, incluyendo una precisión de MVD de píxeles mayores que un número entero (por ejemplo, 2, 3, 4 o mayor precisión de píxeles), y codificar el MVD para el bloque actual de datos de vídeo.

Para calcular valores para posiciones de píxeles subenteros (por ejemplo, cuando se realiza interpredicción), el codificador de vídeo 20 puede incluir una variedad de filtros de interpolación. Por ejemplo, la interpolación bilineal puede usarse para calcular valores para la posición de píxeles subenteros. El codificador de vídeo 20 puede configurarse para realizar la búsqueda de movimiento con respecto a los datos de luminancia de un PU para calcular un vector de movimiento usando los datos de luminancia del PU. El codificador de vídeo 20 puede a continuación reusar el vector de movimiento para codificar los datos de crominancia del PU. Típicamente, los datos de crominancia tienen una resolución más baja que los datos de luminancia correspondientes, p. ej., una cuarta parte de la resolución de los datos de luminancia. Por lo tanto, el vector de movimiento para los datos de crominancia puede tener una precisión más alta que para los datos de luminancia. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar vectores de movimiento de precisión de un cuarto de píxel (y MVD calculados) para datos de luminancia, y pueden usar precisión de un octavo de píxel para datos de crominancia. De manera similar, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden usar vectores de movimiento de precisión de un octavo de píxel para datos de luminancia, y pueden usar precisión de un dieciseisavo de píxel para datos de crominancia.

Después de la codificación intrapredictiva o interpredictiva para producir datos predictivos y datos residuales, y después de cualquier transformada (como la transformada entera 4x4 u 8x8 usada en H.264/AVC o una transformada de coseno discreta DCT) para producir coeficientes de transformada, se puede realizar la cuantificación de coeficientes de transformada. La cuantificación generalmente se refiere a un proceso donde se cuantifican los coeficientes de transformación para reducir la cantidad de datos usados para representarlos. El procedimiento de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. Por ejemplo, un valor de n bits puede redondearse a un valor de m bits durante la cuantificación, donde n es mayor que m .

Después de la cuantificación, se puede realizar codificación entrópica de los datos cuantificados, por ejemplo, según codificación de longitud variable adaptativa de contenido (*Content Adaptive Variable Length Coding*, CAVLC), codificación aritmética binaria adaptativa de contexto (*Context Adaptive Binary Arithmetic Coding*, CABAC) u otra metodología de codificación entrópica. Una unidad de procesamiento configurada para codificación de entropía, u otra unidad de procesamiento, puede realizar otras funciones de procesamiento, tales como codificación de longitud de serie cero de coeficientes cuantificados y/o generación de información sintáctica tal como valores de patrón de bloques codificados (*Coded Block Pattern*, CBP), tipo de macrobloque, modo de codificación, tamaño de LCU o similares.

Como se explicará con más detalle a continuación, el codificador de vídeo 20 puede configurarse para recibir un bloque de datos de vídeo a codificar usando un modo de interpredicción, determinar un MVD para el bloque de datos de vídeo, determinar una precisión de MVD actual, a partir de tres o más precisiones de MVD, para codificar uno o más elementos de sintaxis que indican el MVD, donde las tres o más precisiones de MVD incluyen una precisión de MVD de N muestras, donde N es un número entero que indica una cantidad de muestras indicadas por cada palabra de código sucesiva de los uno o más elementos de sintaxis que indican el MVD, y donde N es mayor que 1, codificar el

bloque de datos de vídeo usando el MVD y el modo de interpredicción, y codificar los uno o más elementos de sintaxis que indican el MVD usando la precisión de MVD actual determinada. Del mismo modo, el decodificador de vídeo 30 puede configurarse para recibir un bloque codificado de datos de vídeo que se codificó usando un modo de interpredicción, recibir uno o más elementos de sintaxis que indican un MVD asociado con el bloque codificado de datos de vídeo, determinar una precisión de MVD actual, a partir de tres o más precisiones de MVD, para el uno o más elementos de sintaxis que indican el MVD, donde las tres o más precisiones de MVD incluyen una precisión de MVD de N muestras, donde N es un número entero que indica una cantidad de muestras indicadas por cada palabra de código sucesiva del uno o más elementos de sintaxis que indican el MVD, y donde N es mayor que 1, decodificar el uno o más elementos de sintaxis que indican el MVD usando la precisión de MVD actual determinada, y decodificar el bloque codificado de datos de vídeo usando el MVD decodificado.

El decodificador de vídeo 30 del dispositivo de destino 14 se puede configurar para realizar técnicas similares, y generalmente simétricas, a cualquiera o todas las técnicas del codificador de vídeo 20 de esta descripción.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar técnicas para codificar y/o señalar vectores de movimiento y/o MVD. El codificador de vídeo 20 puede realizar una intrapredicción e interpredicción de bloques dentro de fotogramas de vídeo, incluidas LCU, CU y PU, y calcular valores residuales que pueden codificarse como TU. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de un fotograma determinado. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de fotogramas adyacentes de una secuencia de vídeo. Intramodo (modo I) puede referirse a cualquiera de varios modos de compresión basados en el espacio e intermodos como la predicción unidireccional (modo P) o la predicción bidireccional (modo B) pueden referirse a cualquiera de varios modos de compresión basados en el tiempo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden realizar codificación de interpredicción, mientras que la unidad de intrapredicción 46 puede realizar codificación de intrapredicción.

Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de un fotograma a codificar. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una memoria de datos de vídeo 38, una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una unidad de intrapredicción 46, un búfer de imagen decodificada 64, un sumador 50, una unidad de transformación 52, una unidad de cuantificación 54 y una unidad de codificación de entropía 56. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye la unidad de cuantificación inversa 58, la unidad de transformación inversa 60 y el sumador 62. En algunos ejemplos, un filtro de eliminación de bloques (no mostrado en la Figura 2) también se incluye para filtrar los límites de bloque para eliminar los artefactos de bloques del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtraría típicamente la salida del sumador 62.

La memoria de datos de vídeo 38 puede almacenar datos de vídeo a codificar por los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 38 se pueden obtener, por ejemplo, a partir de la fuente de vídeo 18. El búfer de imagen decodificada 64 puede ser una memoria de imagen de referencia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por el codificador de vídeo 20, por ejemplo, en modos de intra o intercodificación. La memoria de datos de vídeo 38 y el búfer de imágenes decodificadas 64 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como memoria de acceso aleatorio dinámica (*Dynamic Random Access Memory*, DRAM), lo que incluye una DRAM síncrona (*Synchronous Dynamic Random Access Memory*, SDRAM), una RAM magnetorresistiva (*Magnetoresistive RAM*, MRAM), una RAM resistiva (*Resistive RAM*, RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 38 y el búfer de imágenes decodificadas 64 se pueden proporcionar mediante el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, la memoria de datos de vídeo 38 puede estar incorporada en el chip con otros componentes del codificador de vídeo 20, o no incorporada en el chip relacionado con esos componentes.

Durante el proceso de codificación, el codificador de vídeo 20 recibe un fotograma o fragmento de vídeo que se va a codificar. El fotograma o fragmento puede dividirse en varios bloques de vídeo (p. ej., LCU). La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan la codificación interpredictiva del bloque de vídeo recibido con respecto a uno o más bloques en uno o más fotogramas de referencia para proporcionar compresión temporal. La unidad de intrapredicción 46 puede realizar la codificación intrapredictiva del bloque de vídeo recibido con respecto a uno o más bloques vecinos en el mismo fotograma o fragmento que el bloque que se va a codificar para proporcionar compresión espacial.

La unidad de selección de modo 40 puede seleccionar uno de los modos de predicción, intra o inter, por ejemplo, en base a los resultados de error, y proporciona el bloque predicho resultante al sumador 50 para generar datos residuales y al sumador 62 para reconstruir el bloque codificado para su uso como una imagen de referencia. Cuando la unidad de selección de modo 40 selecciona la codificación entre modos para un bloque, la unidad de selección de resolución 48 puede seleccionar una resolución para un vector de movimiento (por ejemplo, un subpíxel o precisión de píxel

entero) para el bloque. Por ejemplo, la unidad de selección de resolución 48 puede seleccionar una precisión de un octavo de píxel o una precisión de un cuarto de píxel para un vector de movimiento y MVD para el bloque.

Como ejemplo, la unidad de selección de resolución 48 puede configurarse para comparar una diferencia de error (por ejemplo, la diferencia entre un bloque reconstruido y el bloque original) entre el uso de un vector de movimiento de precisión de un cuarto de píxel para codificar un bloque y el uso de un vector de movimiento de precisión de un octavo de píxel para codificar el bloque. La unidad de estimación de movimiento 42 se puede configurar para codificar un bloque usando uno o más vectores de movimiento de precisión de un cuarto de píxel en una primera pasada de codificación y uno o más vectores de movimiento de precisión de un octavo de píxel en una segunda pasada de codificación. La unidad de estimación de movimiento 42 puede usar además una variedad de combinaciones de uno o más vectores de movimiento de precisión de cuarto de píxel y uno o más vectores de movimiento de precisión de octavo píxel para el bloque en una tercera pasada de codificación. La unidad de selección de resolución 48 puede calcular valores de tasa-distorsión para cada etapa de codificación del bloque y calcular diferencias entre los valores de tasa-distorsión.

Cuando la diferencia excede un umbral, la unidad de selección de resolución 48 puede seleccionar el vector de movimiento de precisión de un octavo de píxel para codificar el bloque. La unidad de selección de resolución 48 también puede evaluar la información de velocidad-distorsión, analizar un presupuesto de bits y/o analizar otros factores para determinar si usar una precisión de un octavo de píxel o una precisión de un cuarto de píxel para un vector de movimiento cuando se codifica un bloque durante un proceso de predicción intermodo. Después de seleccionar una precisión de un octavo de píxel o una precisión de un cuarto de píxel para un bloque que se va a codificar intermodo, la unidad de selección de modo 40 o la estimación de movimiento pueden enviar un mensaje (por ejemplo, una señal) a la unidad de estimación de movimiento 42 indicativo de la precisión seleccionada para un vector de movimiento.

Además, según las técnicas de la descripción descritas con más detalle a continuación, el codificador de vídeo 20 puede configurarse para determinar una diferencia de vector de movimiento (MVD) para un bloque actual de datos de vídeo usando una de una pluralidad de precisiones MVD, y codificar el MVD para el bloque actual de datos de vídeo. Como se explicará con más detalle a continuación, el codificador de vídeo 20 puede configurarse para recibir un bloque de datos de vídeo a codificar usando un modo de interpredicción, determinar un MVD para el bloque de datos de vídeo, determinar una precisión de MVD actual, a partir de tres o más precisiones de MVD, para codificar uno o más elementos de sintaxis que indican el MVD, donde las tres o más precisiones de MVD incluyen una precisión de MVD de N muestras, donde N es un número entero que indica una cantidad de muestras indicadas por cada palabra de código sucesiva de los uno o más elementos de sintaxis que indican el MVD, y donde N es mayor que 1, codificar el bloque de datos de vídeo usando el MVD y el modo de interpredicción, y codificar los uno o más elementos de sintaxis que indican el MVD usando la precisión de MVD actual determinada.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento es el proceso de generar vectores de movimiento que estiman el movimiento de los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un bloque predictivo dentro de un fotograma de referencia predictivo (u otra unidad codificada) con respecto al bloque actual que se está codificando dentro del fotograma actual (u otra unidad codificada). Un bloque predictivo es un bloque que coincide en gran medida con el bloque que se va a codificar, en términos de diferencia de píxeles, que puede determinarse mediante la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD) u otras métricas de diferencia. Un vector de movimiento también puede indicar el desplazamiento de una partición de un macrobloque. La compensación de movimiento puede implicar la obtención o generación del bloque predictivo según el vector de movimiento determinado por la estimación de movimiento. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente, en algunos ejemplos.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para el bloque de vídeo de un fotograma intercodificado comparando el bloque de vídeo con bloques de vídeo de un fotograma de referencia en el búfer de imágenes decodificadas 64. La unidad de compensación de movimiento 44 también puede interpolar píxeles subenteros del fotograma de referencia, por ejemplo, un fotograma I o un fotograma P. La norma ITU H.264, como ejemplo, describe dos listas: la lista 0, que incluye fotogramas de referencia que tienen un orden de visualización anterior a un fotograma actual que se codifica, y la lista 1, que incluye fotogramas de referencia que tienen un orden de visualización posterior al fotograma actual que se codifica. Por lo tanto, los datos almacenados en el búfer de imágenes decodificadas 64 pueden organizarse según estas listas.

En algunos ejemplos, la unidad de compensación de movimiento 44 puede configurarse para interpolar valores para posiciones de un decimosexto píxel de datos de crominancia de un CU cuando un vector de movimiento para datos de luminancia de la CU tiene una precisión de un octavo de píxel. Para interpolar valores para las posiciones de un decimosexto píxel de los datos de crominancia, la unidad de compensación de movimiento 44 puede usar interpolación bilineal. Por lo tanto, el sumador 50 puede calcular un residuo para los datos de crominancia de la CU en relación con los valores interpolados bilineales de las posiciones de un decimosexto píxel de un bloque de referencia. De esta

manera, el codificador de vídeo 20 puede calcular, usando interpolación bilineal, valores de posiciones de un decimosexto píxel de datos de crominancia de un bloque de referencia identificado por un vector de movimiento y codificar datos de crominancia de una unidad de codificación según los valores interpolados bilineales del bloque de referencia, cuando los datos de luminancia de la unidad de codificación se codificaron usando un vector de movimiento que tiene una precisión de un octavo de píxel para los datos de luminancia.

La unidad de estimación de movimiento 42 compara bloques de uno o más fotogramas de referencia del búfer de imágenes decodificadas 64 con un bloque a codificar de un fotograma actual, por ejemplo, un fotograma P o un fotograma B. Cuando los fotogramas de referencia en el búfer de imágenes decodificadas 64 incluyen valores para píxeles subenteros, un vector de movimiento calculado por la unidad de estimación de movimiento 42 puede referirse a una ubicación de píxel subentero de un fotograma de referencia. La unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 también pueden configurarse para calcular valores para posiciones de píxeles subenteros de fotogramas de referencia almacenados en el búfer de imágenes decodificadas 64 si no hay valores para posiciones de píxeles subenteros almacenados en el búfer de imágenes decodificadas 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación entrópica 56 y la unidad de compensación de movimiento 44. El bloque de fotograma de referencia identificado por un vector de movimiento puede denominarse un bloque predictivo.

La unidad de estimación de movimiento 42, la unidad de compensación de movimiento 44, la unidad de selección de modo 40 u otra unidad de codificador de vídeo 20, también pueden señalar el uso de una precisión de un cuarto de píxel, una precisión de un octavo de píxel u otra precisión de vector de movimiento (por ejemplo, precisión de píxel entero o precisión de píxel de muestra múltiple) para un vector de movimiento usado para codificar un bloque. Por ejemplo, la unidad de estimación de movimiento 42 puede enviar una indicación de una precisión de píxel subentero para el vector de movimiento a la unidad de codificación entrópica 56. La unidad de estimación de movimiento 42 también puede proporcionar información de contexto relacionada con la información de tamaño para una PU correspondiente al vector de movimiento a la unidad de codificación entrópica 56, donde la información de tamaño puede incluir cualquiera o la totalidad de una profundidad de una CU que incluye la PU, un tamaño de la PU y/o un tipo para la PU.

La unidad de compensación de movimiento 44 puede calcular datos de predicción basándose en el bloque predictivo. El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de compensación de movimiento 44 del bloque de vídeo original que se está codificando. El sumador 50 representa el componente o componentes que realizan esta operación de resta. La unidad de transformación 52 aplica una transformación, tal como una transformada de coseno discreta (DCT) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo comprendiendo valores de coeficientes de transformación residuales.

La unidad de transformación 52 puede realizar otras transformaciones, tales como las definidas por el estándar H.264, que son conceptualmente similares a DCT. Se podrían usar transformaciones de ondículas, transformaciones de enteros, transformaciones de sub-bandas, DST u otros tipos de transformaciones en lugar de una DCT. En cualquier caso, la unidad de transformación 52 aplica la transformación al bloque residual, produciendo un bloque de coeficientes de transformación residuales. La transformación puede convertir la información residual de un dominio de píxeles a un dominio de transformación, tal como un dominio de frecuencia. La unidad de cuantificación 54 cuantifica los coeficientes de transformación residuales para reducir aún más la tasa de bits. El procedimiento de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación.

Después de la cuantificación, la unidad de codificación entrópica 56 codifica por entropía los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar CAVLC de contenido, CABAC u otra técnica de codificación por entropía. Tras la codificación por entropía realizada por la unidad 56, el vídeo codificado puede transmitirse a otro dispositivo o archivarse para su posterior transmisión o recuperación. En el caso de la codificación aritmética binaria adaptativa al contexto, el contexto puede basarse en macrobloques adyacentes.

En algunos casos, la unidad de codificación por entropía 56 u otra unidad del codificador de vídeo 20 puede configurarse para realizar otras funciones de codificación, además de la codificación por entropía. Por ejemplo, la unidad de codificación entrópica 56 puede configurarse para determinar los valores de CBP para los macrobloques y particiones. Además, en algunos casos, la unidad de codificación entrópica 56 puede realizar una codificación de longitud de ejecución de los coeficientes en un macrobloque o partición de este. En particular, la unidad de codificación entrópica 56 puede aplicar un escaneo en zigzag u otro patrón de escaneo para escanear los coeficientes de transformada en un macrobloque o partición y codificar series de ceros para una compresión adicional. La unidad de codificación entrópica 56 también puede construir información de encabezado con elementos sintácticos apropiados para la transmisión en el flujo de bits de vídeo codificado.

Según las técnicas de esta descripción, en los casos en que la precisión de MV/MVD se señala en lugar de derivarse, la unidad de codificación por entropía 56 puede configurarse para codificar una indicación de una precisión de MV/MVD

para un vector de movimiento y/o MVD, por ejemplo, para indicar si el vector de movimiento tiene precisión de píxeles enteros o tiene precisión de subpíxeles, tal como precisión de un cuarto de píxel o precisión de un octavo de píxel (u otras precisiones de subpíxeles, en diversos ejemplos). La unidad de codificación entrópica 56 puede codificar la indicación usando CABAC. Además, la unidad de codificación por entropía 56 puede usar información de contexto para realizar CABAC para codificar la indicación que indica información de tamaño para un PU correspondiente al vector de movimiento, donde la información de tamaño puede incluir cualquiera o la totalidad de una profundidad de una CU que incluye el PU, un tamaño del PU y/o un tipo para el PU.

Tal como se analizó anteriormente, el codificador de vídeo 20 puede señalar de forma predictiva el vector de movimiento. Dos ejemplos de técnicas de señalización predictiva que pueden implementarse mediante el codificador de vídeo 20 incluyen AMVP y señalización de modo de fusión. En AMVP, el codificador de vídeo 20 y el decodificador de vídeo 30 ensamblan listas de candidatos basándose en vectores de movimiento determinados a partir de bloques ya codificados. El codificador de vídeo 20 a continuación señala un índice en la lista de candidatos para identificar un predictor de vector de movimiento (MVP) y señala un MVD. El decodificador de vídeo 30 inter predice un bloque usando el MVP modificado por el MVD, por ejemplo, usando un vector de movimiento igual a MVP + MVD.

En el modo de fusión, el codificador de vídeo 20 y el decodificador de vídeo 30 ensamblan una lista de candidatos basada en bloques ya codificados, y el codificador de vídeo 20 señala un índice para uno de los candidatos en la lista de candidatos. En el modo de fusión, el decodificador de vídeo 30 predice el bloque actual usando el vector de movimiento y el índice de imágenes de referencia del candidato señalado. Tanto en el modo AMVP como en el modo de fusión, el codificador de vídeo 20 y el decodificador de vídeo 30 usan las mismas técnicas de construcción de lista, de modo que la lista usada por el codificador de vídeo 20 al determinar cómo codificar un bloque coincide con la lista usada por el decodificador de vídeo 30 al determinar cómo decodificar el bloque.

La unidad de cuantificación inversa 58 y la unidad de transformación inversa 60 aplican cuantificación y transformación inversas, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso posterior como bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia agregando el bloque residual a un bloque predictivo de uno de los fotogramas del búfer de imagen decodificada 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles subenteros para su uso en la estimación de movimiento. El sumador 62 agrega el bloque residual reconstruido al bloque de predicción compensado de movimiento producido por la unidad de compensación de movimiento 44 para producir un bloque de vídeo reconstruido para almacenamiento en el búfer de imagen decodificada 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 intercodificar un bloque en un fotograma de vídeo posterior.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30, que decodifica una secuencia de vídeo codificada. En el ejemplo de la FIG. 3, el decodificador de vídeo 30 incluye una unidad de decodificación de entropía 70, una unidad de compensación de movimiento 72, una unidad de predicción intra 74, una unidad de cuantificación inversa 76, una unidad de transformación inversa 78, un búfer de imagen decodificada 82 y un sumador 80. El decodificador de vídeo 30 puede, en algunos ejemplos, realizar una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción basados en vectores de movimiento recibidos de la unidad de decodificación entrópica 70.

La memoria de datos de vídeo 68 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado, a decodificar por los componentes del decodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 68 pueden obtenerse, por ejemplo, de un medio legible por ordenador 16, por ejemplo, de una fuente de vídeo local, como una cámara, mediante la comunicación de datos de vídeo por red cableada o inalámbrica, o accediendo a medios físicos de almacenamiento de datos. La memoria de datos de vídeo 68 puede formar un búfer de imágenes codificadas (CPB) que almacena datos de vídeo codificados de un flujo de bits de vídeo codificado. El búfer de imágenes decodificadas 82 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la decodificación de datos de vídeo por el decodificador de vídeo 30, por ejemplo, en modos de intracodificación o intercodificación. La memoria de datos de vídeo 68 y el búfer de imágenes decodificadas 82 pueden estar formados por cualquiera de una variedad de dispositivos de memoria, tales como DRAM, incluyendo SDRAM síncrona, MRAM, RRAM u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 68 y el búfer de imágenes decodificadas 82 se pueden proporcionar mediante el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, la memoria de datos de vídeo 68 puede estar en el chip con otros componentes del decodificador de vídeo 30, o fuera del chip con respecto a esos componentes.

La unidad de decodificación por entropía 70 puede recuperar un flujo de bits codificado, por ejemplo, de la memoria de datos de vídeo 68. El flujo de bits codificado puede incluir datos de vídeo codificados por entropía (por ejemplo, bloques codificados de datos de vídeo). La unidad de decodificación por entropía 70 puede decodificar los datos de vídeo codificados por entropía, y a partir de los datos de vídeo decodificados por entropía, la unidad de compensación de movimiento 72 puede determinar información de movimiento, incluyendo vectores de movimiento, precisión de

vector de movimiento, índices de lista de imágenes de referencia y otra información de movimiento. La unidad de compensación de movimiento 72 puede, por ejemplo, determinar dicha información realizando las técnicas de AMVP y modo de fusión descritas anteriormente.

La unidad de compensación de movimiento 72 puede usar vectores de movimiento y/o MVD recibidos en el flujo de bits para identificar un bloque de predicción en fotogramas de referencia en el búfer de imágenes decodificadas 82. La precisión usada para codificar vectores de movimiento y/o MVD puede definirse mediante indicadores de precisión (por ejemplo, uno o más elementos sintácticos) que son decodificados por la unidad de decodificación de entropía 70. La unidad de intrapredicción 74 puede usar modos de intrapredicción recibidos en el flujo de bits para formar un bloque de predicción a partir de bloques espacialmente adyacentes. La unidad de cuantificación inversa 76 cuantifica inversamente, es decir, descuantifica, los coeficientes de bloque cuantificados proporcionados en el flujo de bits y decodificados por la unidad de decodificación entrópica 70. El proceso de cuantificación inversa puede incluir un proceso convencional, por ejemplo, como se define por el estándar de decodificación H.264. El proceso de cuantificación inversa también puede incluir el uso de un parámetro de cuantificación QP_y, calculado por el codificador de vídeo 20 para cada macrobloque, a fin de determinar el grado de cuantificación y, asimismo, el grado de cuantificación inversa que debe aplicarse.

La unidad de transformación inversa 58 aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación entera inversa o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación para producir bloques residuales en el dominio de píxeles. La unidad de compensación de movimiento 72 produce bloques compensados de movimiento, posiblemente realizando interpolación basada en filtros de interpolación. Los identificadores de los filtros de interpolación que se usan para la estimación de movimiento con precisión de subpíxeles pueden incluirse en los elementos de sintaxis.

La unidad de compensación de movimiento 72 puede usar filtros de interpolación similares a los que usa el codificador de vídeo 20 durante la codificación del bloque de vídeo para calcular valores interpolados para píxeles subenteros de un bloque de referencia. La unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 según la información de sintaxis recibida y usarlos para generar bloques predictivos. Según las técnicas de la presente descripción, la unidad de compensación de movimiento 72 puede interpolar valores de posiciones de un decimosexto píxel de datos de crominancia de un bloque de referencia cuando un vector de movimiento tiene una precisión de un octavo de píxel para datos de luminancia. Por ejemplo, la unidad de compensación de movimiento 72 puede usar interpolación bilineal para interpolar los valores de las posiciones de un decimosexto píxel del bloque de referencia.

La unidad de compensación de movimiento 72 usa parte de la información sintáctica para determinar los tamaños de las LCU y las CU usadas para codificar fotogramas y/o fragmentos de la secuencia de vídeo codificada, información de partición que describe cómo se divide cada macrobloque de un fotograma de la secuencia de vídeo codificada, modos que indican cómo se codifica cada partición, uno o más fotogramas de referencia (y listas de fotogramas de referencia) para cada CU intercodificada y otra información para decodificar la secuencia de vídeo codificada.

Según las técnicas de la descripción descritas con más detalle a continuación, el decodificador de vídeo 30 puede configurarse para determinar un MVD para un bloque actual de datos de vídeo usando una de una pluralidad de precisiones de MVD, y codificar el MVD para el bloque actual de datos de vídeo. Como se explicará con más detalle a continuación, el decodificador de vídeo 30 se puede configurar para recibir un bloque codificado de datos de vídeo que se codificó usando un modo de interpretación, recibir uno o más elementos de sintaxis que indican un MVD asociado con el bloque codificado de datos de vídeo, determinar una precisión de MVD actual, a partir de tres o más precisiones de MVD, para el uno o más elementos de sintaxis que indican el MVD, donde las tres o más precisiones de MVD incluyen una precisión de MVD de N muestras, donde N es un número entero que indica una cantidad de muestras indicadas por cada palabra de código sucesiva del uno o más elementos de sintaxis que indican el MVD, y donde N es mayor que 1, decodificar el uno o más elementos de sintaxis que indican el MVD usando la precisión de MVD actual determinada, y decodificar el bloque codificado de datos de vídeo usando el MVD decodificado.

El sumador 80 suma los bloques residuales con los bloques de predicción correspondientes generados por la unidad de compensación de movimiento 72 o la unidad de intrapredicción para formar bloques decodificados. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques decodificados con el fin de eliminar los artefactos de bloques. Los bloques de vídeo decodificados se almacenan luego en el búfer de imágenes decodificadas 82, que proporciona bloques de referencia para la compensación de movimiento posterior y también produce vídeo decodificado para su presentación en un dispositivo de visualización (tal como el dispositivo de visualización 32 de la FIG. 1).

La FIG. 4 es un diagrama conceptual que ilustra posiciones de píxel fraccionarias para una posición de píxel completa. En particular, la FIG. 4 ilustra posiciones de píxeles fraccionarios para píxeles completos (pel) 100. El píxel completo 100 corresponde a las posiciones de medio píxel 102A-102C (medio píxel 102), las posiciones de cuarto de píxel 104A-104L (cuarto de píxel 104) y las posiciones de un octavo de píxel 106A-106AV (octavo de píxel 106).

La FIG. 4 ilustra las posiciones del octavo píxel 106 de un bloque usando un contorno discontinuo para indicar que estas posiciones pueden incluirse opcionalmente. Es decir, si un vector de movimiento (por ejemplo, como se reconstruye a partir de un MVD de la misma precisión) tiene una precisión de un octavo de píxel, el vector de movimiento puede apuntar a cualquiera de la posición de píxel completa 100, las posiciones de medio píxel 102, las posiciones de cuarto de píxel 104 o las posiciones de octavo de píxel 106. Sin embargo, si el vector de movimiento tiene una precisión de un cuarto de píxel, el vector de movimiento puede apuntar a cualquiera de la posición de píxel completa 100, las posiciones de medio píxel 102 o las posiciones de cuarto de píxel 104, pero no apuntaría a las posiciones de octavo píxel 106. Además, debe entenderse que, en otros ejemplos, se pueden usar otras precisiones, por ejemplo, una decimosexta precisión de píxel, una precisión de píxel de treinta segundos o similares.

Se puede incluir un valor para el píxel en la posición de píxel completa 100 en un fotograma de referencia correspondiente. Es decir, el valor para el píxel en la posición de píxel completa 100 generalmente corresponde al valor real de un píxel en el fotograma de referencia, por ejemplo, que finalmente se representa y se muestra cuando se muestra el fotograma de referencia. Los valores para las posiciones de medio píxel 102, las posiciones de cuarto de píxel 104 y las posiciones de octavo píxel 106 (denominadas colectivamente posiciones de píxel fraccionarias) pueden interpolarse usando filtros de interpolación adaptativos o filtros de interpolación fijos, por ejemplo, filtros de varios números de "derivaciones" (coeficientes), tales como varios filtros de Wiener, filtros bilineales u otros filtros. En general, el valor de una posición de píxel fraccional puede interpolarse a partir de uno o más píxeles vecinos, que corresponden a valores de posiciones de píxel completas vecinas o posiciones de píxel fraccional determinadas previamente.

En algunos ejemplos de esta descripción, como se tratará con más detalle a continuación, el codificador de vídeo 20 puede seleccionar de forma adaptativa la precisión para un vector de movimiento y/o MVD, por ejemplo, entre la precisión de píxeles de múltiples muestras, la precisión de píxeles enteros o la precisión de píxeles fraccionarios, como la precisión de un octavo de píxel y la precisión de un cuarto de píxel. El codificador de vídeo 20 puede realizar esta selección para cada vector de movimiento, cada CU, cada LCU, cada fragmento, cada fotograma, cada GOP u otras unidades codificadas de datos de vídeo. Cuando el codificador de vídeo 20 selecciona una precisión de un cuarto de píxel para un vector de movimiento, el vector de movimiento puede referirse a cualquiera de la posición de píxel completa 100, las posiciones de medio píxel 102 o las posiciones de cuarto de píxel 104. Cuando el codificador de vídeo 20 selecciona una precisión de un octavo de píxel para un vector de movimiento, el vector de movimiento puede referirse a cualquiera de la posición de píxel completa 100, las posiciones de medio píxel 102, las posiciones de cuarto de píxel 104 o las posiciones de octavo de píxel 106.

Las FIGS. 5A-5C son diagramas conceptuales que ilustran las posiciones de píxeles de crominancia y luminancia correspondientes. Las FIGS. 5A-5C también ilustran cómo los vectores de movimiento calculados para los datos de luminancia pueden reusarse para los datos de crominancia. Como cuestión preliminar, las FIGS. 5A-5C ilustran una fila parcial de posiciones de píxeles. Se debe entender que, en la práctica, una posición de píxel completa puede tener una cuadrícula rectangular de posiciones de píxel fraccionarias asociadas, tal como la ilustrada en la FIG. 4. El ejemplo de las FIGS. 5A-5C pretenden ilustrar los conceptos descritos en esta descripción y no pretenden ser una lista exhaustiva de correspondencias entre las posiciones de píxeles de crominancia fraccional y las posiciones de píxeles de luminancia fraccional.

Las FIGS. 5A-5C ilustran posiciones de píxel de un bloque de luminancia, que incluye la posición de píxel de luminancia completa 110, la posición de píxel de media luminancia 116, la posición de cuarto de píxel 112 y las posiciones de octavo píxel de luminancia 114A, 114B. Las FIGS. 5A-5C también ilustran las posiciones de píxel correspondientes del bloque de crominancia, incluyendo la posición de píxel de crominancia completa 120, la posición de píxel de crominancia de cuarto 122, la posición de píxel de crominancia de octavo 124 y las posiciones de píxel de crominancia de decimosexto 126A, 126B. En este ejemplo, el píxel de crominancia completa 120 corresponde al píxel de luminancia completa 110. Además, en este ejemplo, el bloque de crominancia se submuestra en un factor de dos horizontal y verticalmente con respecto al bloque de luminancia. Por lo tanto, el píxel de cuarto de crominancia 122 corresponde al píxel de mitad de luminancia 116. De manera similar, el octavo píxel de crominancia 124 corresponde al cuarto de píxel de luminancia 112, el decimosexto píxel de crominancia 126A corresponde al octavo píxel de luminancia 114A y el decimosexto píxel de crominancia 126B corresponde al octavo píxel de luminancia 114B.

En códecs de vídeo avanzados, como H.264/AVC, H.265/HEVC, y potencialmente el éxito o los códecs a H.264 y H.265/HEVC, el coste de bits de los vectores de movimiento de señalización puede aumentar. Para reducir este coste de bits, se puede usar la derivación MV del lado del decodificador (*Decoder side MV Derivation, DMVD*). En S. Kamp y M. Wien, "Decoder-side motion vector derivation for block-based video coding", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, págs. 1732 -1745, diciembre de 2012, se propuso DMVD según una coincidencia de plantillas en forma de L.

La FIG. 6 es una ilustración de un ejemplo de plantilla en forma de L que coincide con DMVD. En el ejemplo de la FIG. 6, el bloque actual 132 de la imagen actual 134 se interpredice mediante coincidencia de plantillas. La plantilla 136 define una forma que cubre bloques vecinos ya decodificados del bloque actual 132. Un decodificador de vídeo (por ejemplo, el decodificador de vídeo 30) puede, por ejemplo, comparar primero los valores de píxel incluidos en los

bloques vecinos ya decodificados cubiertos por la plantilla 136 con los valores de píxel incluidos en los bloques vecinos ya decodificados cubiertos por la plantilla 138, que cubre los bloques ubicados en una imagen de referencia de las imágenes de referencia 140. El decodificador de vídeo puede mover la plantilla a otras ubicaciones en la imagen de referencia y comparar los valores de píxeles cubiertos por la plantilla con los valores de píxeles incluidos en los bloques vecinos ya decodificados cubiertos por la plantilla 136.

Con base en estas comparaciones múltiples, el decodificador de vídeo puede determinar una mejor coincidencia, tal como la mejor coincidencia 142 mostrada en el ejemplo de la FIG. 6. El decodificador de vídeo puede entonces determinar un desplazamiento entre la mejor coincidencia y la plantilla colocada. Este desplazamiento (por ejemplo, desplazamiento 144 en la FIG. 6) corresponde al vector de movimiento usado para predecir el bloque actual 132.

Como se ilustra en la FIG. 6, cuando un bloque se codifica en modo DMVD, el MV para el bloque se busca mediante el decodificador de vídeo 30, en lugar de señalizarse directamente al decodificador de vídeo 30. El MV que conduce a la distorsión mínima mediante la coincidencia de plantillas se selecciona como el MV final para el bloque. Para mantener una alta eficiencia de codificación, puede ser necesario un cierto número de coincidencias de plantillas para que el decodificador 30 seleccione un vector de movimiento candidato como el MV para decodificar el bloque actual, lo que puede aumentar la complejidad de decodificación.

Para reducir la complejidad de la decodificación en DMVD, se propuso un procedimiento de derivación de MV bidireccional basado en espejo en el documento Y.-J. Chiu, L. Xu, W. Zhang, H. Jiang, "DECODER-SIDE MOTION ESTIMATION AND WIENER FILTER FOR HEVC", taller VCIP 2013, Malasia, 17-20 de noviembre de 2013.

La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de derivación de MV bidireccional basada en espejo. Como se ilustra en la FIG. 7, la derivación de MV bidireccional basada en espejo puede aplicarse mediante la estimación de movimiento centro-simétrico alrededor de los centros de búsqueda en la precisión de la muestra fraccional en el lado del decodificador. El tamaño/ubicación de la ventana de búsqueda puede estar predefinido y señalado en el flujo de bits. En la FIG. 7, dMV es un desplazamiento que se suma a PMV0 y se resta de PMV1 para generar un par de MV, MV0 y MV1. Se pueden verificar todos los valores de dMV dentro de una ventana de búsqueda y se puede usar la Suma de Diferencia Absoluta (SAD) entre los bloques de referencia L0 y de referencia L1 como la medición de la estimación de movimiento centro-simétrico. Se puede seleccionar un par de MV con un SAD mínimo como los MV finales para el bloque.

A continuación, se analizarán las técnicas relacionadas con la resolución adaptativa del vector de movimiento. La compensación de movimiento de subpíxeles a menudo es más eficiente que la compensación de movimiento de píxeles enteros. Sin embargo, para algunos contenidos de vídeo, como el contenido de textura con componentes de muy alta frecuencia o el contenido de la pantalla, la compensación de movimiento de subpíxeles no muestra un rendimiento mejor o incluso peor. En tales casos, puede ser mejor tener solo MV y/o MVD codificados con precisión de píxeles enteros.

Como se describe en el documento L. Guo, P. Yin, Y. Zheng, X. Lu, Q. Xu, J. Sole, "Adaptive motion vector resolution with implicit signaling", ICIP2010: 2057-2060, la resolución de MV adaptativa se describió según los residuos reconstruidos (por ejemplo, valores residuales). Cuando la varianza del bloque de residuos reconstruido está por encima de un umbral, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden determinar el uso de la precisión del vector de movimiento de un cuarto de píxel. De lo contrario, el codificador de vídeo 20 y el decodificador de vídeo 30 usan una precisión de vector de movimiento de medio píxel cuando codifican vectores de movimiento y MVD.

Como se describe en el documento J. An, X. Li, X. Guo, S. Lei, "Progressive MV Resolution", JCTVC-F125, Torino, Italia, julio de 2011, la resolución de MV se determina adaptativamente según la magnitud de la diferencia de MV señalada. Como se describe en el documento Y. Zhou, B. Li, J. Xu, G. J. Sullivan, B. Lin, "Motion Vector Resolution Control for Screen Content Coding", JCTVC-P0277, San José, EE. UU., enero de 2014, en algunos ejemplos, el codificador de vídeo 20 puede configurarse para señalar información de precisión del vector de movimiento a nivel de fragmento.

En otros ejemplos, tales como los descritos en la publicación de patente estadounidense n. ° 2015/0195562, el codificador de vídeo 20 puede configurarse para señalar condicionalmente la información de precisión de MV usando una o más de las siguientes técnicas. Como se analizó anteriormente, en algunos ejemplos, se propone un procedimiento de derivación de precisión del vector de movimiento del lado del decodificador para el contenido de la pantalla. En este ejemplo, la precisión del vector de movimiento puede depender del resultado de la coincidencia de plantillas en el decodificador de vídeo 30. Cuando un resultado de coincidencia de plantilla de una posición de píxel entero y el de su posición de subpíxel vecino son bastante diferentes (por ejemplo, mayores que alguna distancia umbral), la región relacionada puede considerarse como contenido de pantalla y el decodificador de vídeo 30 puede decodificar MV con precisión de píxel entero. De lo contrario, el decodificador de vídeo 30 puede configurarse para usar precisión de vector de movimiento de subpíxel. Para definir "bastante diferente", se pueden usar uno o más umbrales fijos o adaptativos.

El decodificador de vídeo 30 puede, por ejemplo, decodificar datos de vídeo determinando una precisión del vector de movimiento según la coincidencia de plantillas. En dicho ejemplo, el decodificador de vídeo 30 puede, para un bloque actual que se está codificando, identificar una posición de píxel entero de un bloque vecino ya codificado y, según una ubicación de la posición de píxel entero, aplicar una plantilla para determinar una pluralidad de posiciones de píxel entero. El decodificador de vídeo 30 también puede aplicar la plantilla a una pluralidad de posiciones de subpíxel para determinar una pluralidad de posiciones de subpíxel. La plantilla puede, por ejemplo, definir una forma, y el decodificador de vídeo 30 puede aplicar la plantilla a los datos de vídeo para determinar la pluralidad de posiciones de píxeles enteros ubicando la pluralidad de posiciones de píxeles enteros según una ubicación de la forma con respecto al bloque actual. De manera similar, el decodificador de vídeo 30 puede aplicar la plantilla a los datos de vídeo para determinar la pluralidad de posiciones de subpíxeles ubicando la pluralidad de posiciones de píxeles de subpíxeles según una ubicación de la forma con respecto al bloque actual.

El decodificador de vídeo 30 puede comparar uno o más valores de píxeles para la pluralidad de posiciones de píxeles enteros con uno o más valores de píxeles para la pluralidad de posiciones de subpíxeles y, según la comparación, determinar una precisión del vector de movimiento para un vector de movimiento. El decodificador de vídeo 30 puede decodificar el bloque actual usando el vector de movimiento. El decodificador de vídeo 30 puede, por ejemplo, determinar el vector de movimiento usando un modo de fusión, un modo AMVP o algún otro modo de este tipo.

El decodificador de vídeo 30 puede determinar la precisión del vector de movimiento para el vector de movimiento comparando uno o más valores de píxel para la pluralidad de posiciones de píxel enteras con uno o más valores de píxel para la pluralidad de posiciones de subpíxel para determinar un valor de diferencia que corresponde a una cantidad de diferencia en los valores de píxel entre el uno o más valores de píxel para la pluralidad de posiciones de píxel enteras y el uno o más valores de píxel para la pluralidad de posiciones de subpíxel. En respuesta a que el valor de diferencia es mayor que un valor umbral, el decodificador de vídeo 30 puede determinar que la precisión del vector de movimiento es una precisión de píxel entero. En respuesta a que el valor de diferencia sea menor que un valor umbral, el decodificador de vídeo 30 puede determinar que la precisión del vector de movimiento es una precisión de subpíxel. El valor umbral puede ser un valor fijo, un valor adaptativo o algún otro tipo de valor. Para comparar el uno o más valores de píxel para la pluralidad de posiciones de píxel enteras con el uno o más valores de píxel para la pluralidad de posiciones de subpíxel, el decodificador de vídeo 30 puede, por ejemplo, determinar una suma de diferencias absolutas entre el uno o más valores de píxel para la pluralidad de posiciones de píxel enteras y el uno o más valores de píxel para la pluralidad de posiciones de subpíxel.

Según otras técnicas de esta descripción, la precisión del vector de movimiento puede depender de las propiedades (tales como nitidez, gradiente o si se omite la transformada) de los bloques espacialmente vecinos, los bloques temporalmente vecinos o ambos. La información de precisión del vector de movimiento se puede derivar del decodificador de vídeo 30. Como alternativa o adicionalmente, la precisión del vector de movimiento puede depender de la precisión del vector de movimiento de bloques espacialmente vecinos, bloques temporalmente vecinos o ambos.

El decodificador de vídeo 30 puede, por ejemplo, determinar una precisión del vector de movimiento según las propiedades del bloque vecino. Los bloques vecinos pueden, por ejemplo, incluir al menos un bloque espacialmente vecino y/o al menos un bloque temporalmente vecino. Para un bloque actual que se está codificando, el decodificador de vídeo 30 puede localizar uno o más bloques vecinos y determinar una propiedad del uno o más bloques vecinos. La propiedad puede ser, por ejemplo, uno o más de una nitidez del uno o más bloques vecinos, un gradiente del uno o más bloques vecinos, si uno o más bloques vecinos se codificaron en un modo de omisión, y/o una precisión de vector de movimiento del uno o más bloques vecinos. Según la propiedad del uno o más bloques vecinos, el decodificador de vídeo 30 puede determinar una precisión del vector de movimiento para un vector de movimiento y decodificar el bloque actual usando el vector de movimiento. El decodificador de vídeo 30 puede, por ejemplo, determinar sin señalización (por ejemplo, según un contexto) qué propiedad o propiedades determinar, siempre puede determinar una propiedad o propiedades fijas, o puede recibir una indicación de qué propiedad o propiedades determinar.

En otra técnica de ejemplo de esta descripción, el codificador de vídeo 20 puede señalar un indicador en el flujo de bits (por ejemplo, uno o más elementos sintácticos) que especifica qué procedimiento o procedimientos de precisión del vector de movimiento del lado del decodificador se usan. Por ejemplo, el codificador de vídeo 20 puede señalar el indicador en el flujo de bits directamente y/o el decodificador de vídeo 30 puede derivar el valor del indicador de otra información codificada en el flujo de bits, tal como el tipo de fragmento y el nivel temporal.

El decodificador de vídeo 30 puede, por ejemplo, recibir en un flujo de bits de vídeo codificado, una indicación de un tipo de señalización de precisión de vector de movimiento y, según el tipo de señalización de precisión de vector de movimiento, determinar una precisión de vector de movimiento para un bloque de datos de vídeo. El decodificador de vídeo 30 puede usar un vector de movimiento de la precisión del vector de movimiento determinada para localizar un bloque de referencia para el bloque de datos de vídeo. El tipo de señalización de precisión de vector de movimiento puede ser, por ejemplo, uno de (1) un tipo de coincidencia de plantilla como se describió anteriormente, (2) un tipo basado en propiedades de bloque vecino como se describió anteriormente, o (3) un tipo de señalización directa, como se describirá con más detalle a continuación.

El decodificador de vídeo 30 puede, por ejemplo, recibir la indicación de precisión del vector de movimiento en un encabezado de fragmento, un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o en algún otro nivel. En un ejemplo, el decodificador de vídeo 30 puede recibir la indicación de predicción de vector de movimiento para una o más de una unidad de codificación o una unidad de predicción. La indicación puede, por ejemplo, ser un tipo de fragmento. En otras palabras, el decodificador de vídeo 30 puede determinar un tipo de fragmento para un fragmento particular y, según ese tipo de fragmento, puede determinar una precisión de vector de movimiento para usar para decodificar bloques de ese fragmento. La indicación puede, por ejemplo, ser un nivel temporal de un fragmento. En otras palabras, el decodificador de vídeo 30 puede determinar un nivel temporal para un fragmento y, según el nivel temporal del fragmento, determinar una precisión del vector de movimiento que se usará para los bloques de decodificación del fragmento.

En otro ejemplo, el codificador de vídeo 20 puede configurarse para señalar la información de precisión del vector de movimiento en el flujo de bits, tal como en el nivel de unidad de codificación LCU más grande, el nivel de CU o el nivel de PU. En otras palabras, el codificador de vídeo 20 puede generar uno o más elementos sintácticos para su inclusión en el flujo de bits de datos de vídeo codificados, y el decodificador de vídeo 30 puede analizar esos elementos sintácticos para determinar la precisión del vector de movimiento para un bloque particular de datos de vídeo. Cuando se indica que una CU tiene MV de precisión entera, todas las PU dentro de esta CU tienen precisión de vector de movimiento entero.

En un ejemplo, para el modo de fusión/salto, el decodificador de vídeo 30 puede redondear un vector de movimiento a una precisión entera solo cuando se realiza la compensación de movimiento. El MV no redondeado se puede guardar para la predicción MV de bloques posteriores. Por ejemplo, el decodificador de vídeo 30 puede determinar que un modo de codificación para un primer bloque es un modo de fusión o un modo de omisión y determinar que una precisión de vector de movimiento para el primer bloque es una precisión de píxel entero. El decodificador de vídeo 30 puede construir una lista de candidatos de fusión para el primer bloque que incluye al menos un candidato de vector de movimiento de precisión fraccional. El decodificador de vídeo 30 puede seleccionar el candidato de vector de movimiento de precisión fraccional para decodificar el primer bloque y redondear el candidato de vector de movimiento de precisión fraccional para determinar un vector de movimiento de precisión de píxel entero. El decodificador de vídeo 30 puede localizar un bloque de referencia para el primer bloque usando el vector de movimiento de precisión de píxeles enteros.

Para un segundo bloque (por ejemplo, un bloque codificado según la información del primer bloque), el decodificador de vídeo 30 puede añadir el candidato de vector de movimiento de precisión entera a una lista de candidatos (por ejemplo, una lista de candidatos de fusión o una lista de candidatos de AMVP) para el segundo bloque. En otros ejemplos, sin embargo, el decodificador de vídeo 30 puede añadir el candidato de vector de movimiento de precisión fraccional a una lista de candidatos para un segundo bloque.

Para el intermodo sin fusión/omisión, los predictores de MV pueden redondearse a una precisión de número entero, y el MVD puede señalizarse con precisión de número entero para que el MV redondeado pueda guardarse para la predicción de MV de bloque posterior. Alternativa o adicionalmente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para guardar un MV, antes del redondeo, para bloquear posteriormente la predicción de MV. En un ejemplo, para este caso, el redondeo se puede realizar solo para compensación de movimiento. De manera alternativa o adicional, se puede usar un MV redondeado en la compensación de movimiento y se puede guardar para la predicción de MV de bloque posterior.

Por ejemplo, el decodificador de vídeo 30 puede determinar que un modo de codificación para un primer bloque es distinto de un modo de fusión y determinar que una precisión de vector de movimiento para el primer bloque es una precisión de píxel entero. El decodificador de vídeo 30 puede determinar un MVP de precisión fraccional para el primer bloque y redondear el MVP de precisión fraccional para determinar un MVP de precisión de píxel entero para el primer bloque. El decodificador de vídeo 30 puede determinar un MVD para el primer bloque que es una precisión de píxeles enteros. El decodificador de vídeo 30 puede determinar un vector de movimiento de precisión de píxeles enteros según el MVP de precisión de píxeles enteros y el MVD de precisión fraccional. El decodificador de vídeo 30 puede localizar un bloque de referencia para el primer bloque usando el vector de movimiento de precisión de píxeles enteros.

El decodificador de vídeo 30 puede, por ejemplo, determinar el MVP de precisión fraccional para el primer bloque construyendo una lista de candidatos de AMVP para el primer bloque. La lista de candidatos de AMVP puede incluir un candidato de vector de movimiento de precisión fraccional. El decodificador de vídeo 30 puede seleccionar el candidato de vector de movimiento de precisión fraccional como el MVP de precisión fraccional para el primer bloque. El decodificador de vídeo 30 puede añadir el candidato a vector de movimiento de precisión fraccional a una lista de candidatos para un segundo bloque que se va a predecir usando la información del primer bloque.

De manera alternativa o adicional, en un ejemplo, se puede señalar información de precisión de MVD, y siempre se puede usar MV de precisión de subpíxeles, en algunos ejemplos. La precisión del MVD se puede señalar a nivel de la LCU, a nivel de la CU o a nivel de la PU. En un ejemplo, cuando se indica que un PU (o CU) tiene MVD de precisión

entera, el PU (o todos los Pus dentro de esta CU) pueden tener precisión MVD entera. Para las PU codificadas AMVP, la MVD de las PU puede tener una precisión de píxeles enteros, mientras que la MV predicha y la MV de la PU pueden tener una precisión de subpíxeles. Por lo tanto, la adición de un MVD de precisión entera a un MVP de precisión de subpíxel da como resultado un vector de movimiento de subpíxel.

Por ejemplo, el decodificador de vídeo 30 puede determinar que una precisión MVD para un primer bloque es una precisión de píxel entero. El decodificador de vídeo 30 puede construir una lista de candidatos (por ejemplo, una lista de candidatos de AMVP) de MVP para el primer bloque que incluye al menos un candidato de vector de movimiento de precisión fraccional. El decodificador de vídeo 30 puede seleccionar de la lista de candidatos el candidato de vector de movimiento de precisión fraccional y determinar un vector de movimiento de precisión de píxel fraccional según el candidato de vector de movimiento de precisión fraccional y el MVD de precisión de píxel entero. El decodificador de vídeo 30 puede localizar un bloque de referencia para el primer bloque usando el vector de movimiento de precisión de píxeles fraccionarios.

En otro ejemplo, el indicador de precisión del vector de movimiento se puede aplicar parcialmente a una LCU o una CU. Por ejemplo, el indicador de precisión de enteros de CU no se aplica a sus Pus que se codifican con modos de codificación predefinidos, tales como fusión y omisión, o con particiones predefinidas, tales como particiones no $2N \times 2N$, o con una herramienta de codificación especial, tal como omisión de transformada o sin residuos.

En un ejemplo, el decodificador de vídeo 30 puede determinar una precisión de vector de movimiento predeterminada para datos de vídeo y, en respuesta a un PU de los datos de vídeo que se codifican en un modo especial, ubicar un bloque de referencia para el PU usando un vector de movimiento de la precisión de vector de movimiento predeterminada. El modo especial puede ser, por ejemplo, uno o más de un modo de omisión, un modo de fusión $2N \times 2N$, un modo de fusión, un modo de omisión de transformada o un modo de partición asimétrica. En respuesta a un segundo PU de los datos de vídeo que se codifican usando modos distintos de un modo especial, el decodificador de vídeo 30 puede determinar para el segundo PU de los datos de vídeo, una precisión del vector de movimiento señalado y localizar un bloque de referencia para el segundo PU usando un vector de movimiento de la precisión del vector de movimiento señalado. El decodificador de vídeo 30 puede determinar, para una CU de los datos de vídeo, una precisión de vector de movimiento señalizada que es diferente de la precisión de vector de movimiento predeterminada. La CU puede, por ejemplo, incluir la PU y/o la segunda PU. En un ejemplo, la precisión del vector de movimiento señalado puede ser una precisión de píxel entero, mientras que la precisión del vector de movimiento por defecto es una precisión del vector de movimiento fraccional. En otros ejemplos, la precisión del vector de movimiento por defecto puede ser una precisión del vector de movimiento fraccional.

En un ejemplo, el codificador de vídeo 20 puede configurarse para codificar y señalar información de precisión de MV/MVD solo para la PU o CU que tiene un MVD distinto de cero. Cuando no se señala la información de precisión de MV/MVD, el decodificador de vídeo 30 puede configurarse para usar la precisión de MV de subpíxeles para la PU o CU. El codificador de vídeo 20 puede configurarse para señalar información de precisión de MV/MVD después de codificar y señalar el MVD de una PU o CU. En algunos ejemplos, un MVD igual a cero puede indicar que tanto el componente vertical del MVD como los componentes horizontales del MVD son iguales a 0.

Como un ejemplo, para un bloque actual de datos de vídeo, el decodificador de vídeo 30 puede recibir uno o más elementos sintácticos que indican el valor de MVD. En respuesta a que el valor de MVD es igual a cero, el decodificador de vídeo 30 puede configurarse para determinar que un vector de movimiento para el bloque actual tiene precisión de vector de movimiento de subpíxel. El valor de MVD que es igual a cero puede indicar que tanto un componente x del valor de MVD como un componente y del valor de MVD son iguales a cero.

Para un segundo bloque actual de datos de vídeo, el decodificador de vídeo 30 puede recibir uno o más elementos sintácticos que indican un segundo valor de MVD y, en respuesta a que el segundo valor de MVD es un valor distinto de cero, recibir una indicación de una precisión de vector de movimiento o un segundo vector de movimiento para el segundo bloque actual. El decodificador de vídeo 30 puede localizar, en una imagen de referencia, un bloque de referencia para el segundo bloque actual usando el segundo vector de movimiento. Para el segundo bloque actual, el decodificador de vídeo 30 puede recibir la indicación de la precisión del vector de movimiento después de recibir el segundo valor de MVD.

Cuando la información de precisión de MV/MVD se señala a nivel de PU, el codificador de vídeo 20 puede configurarse para no señalar la información de precisión de MV/MVD si una o más (por ejemplo, cualquiera) de las siguientes condiciones son verdaderas: (1) la PU se codifica con el modo de fusión/omisión, (2) la PU se codifica con el modo AMVP, y MVD en cada dirección de predicción de la PU es igual a cero, o (3) alternativa o adicionalmente, si una CU podría contener Pus intracodificado y Pus intercodificado juntos, lo cual no está permitido en HEVC. Cuando la PU está intracodificada, se omite la señalización de la información de precisión de MV/MVD a nivel de PU.

El decodificador de vídeo 30 puede, por ejemplo, recibir, para un primer bloque de datos de vídeo (por ejemplo, una primera PU), uno o más elementos sintácticos que indican la primera información de precisión del vector de movimiento. En respuesta a un segundo bloque de datos de vídeo que cumple una condición, el decodificador de

vídeo 30 puede determinar que la segunda información del vector de movimiento corresponde a una precisión del vector de movimiento por defecto. En un ejemplo, la condición puede ser que el segundo bloque se codifique usando el modo de fusión o el modo de omisión. En otro ejemplo, la condición puede ser que el segundo bloque se codifique usando el modo AMVP y que un MVD para cada dirección de predicción del segundo bloque sea igual a cero. La precisión predeterminada puede ser, por ejemplo, una precisión fraccionaria en algunos ejemplos o una precisión entera en otros ejemplos. La primera y segunda información de precisión del vector de movimiento pueden, por ejemplo, ser una o ambas de una precisión del vector de movimiento o una precisión MVD.

Cuando la información de precisión de MV/MVD se señala a nivel de CU, la información de precisión de MV/MVD puede no señalizarse si una (y posiblemente una o más) de las siguientes condiciones es verdadera para todos los Pus dentro de la CU: (1) la PU está intracodificada, (2) la PU está codificada con modo de fusión/omisión, o (3) la PU está codificada con modo AMVP, y MVD en cada dirección de predicción de la PU es igual a cero. De forma alternativa o adicional, cuando no se señala la información de precisión del vector de movimiento, se puede usar una precisión del vector de movimiento por defecto, tal como una precisión del vector de movimiento entero, para la PU o la CU.

El decodificador de vídeo 30 puede, por ejemplo, recibir, para una primera CU de datos de vídeo, primera información de precisión de vector de movimiento y, en respuesta a una segunda CU de los datos de vídeo que cumple una condición, determinar que la segunda información de vector de movimiento corresponde a una precisión predeterminada. La condición puede, por ejemplo, ser que todos los Pus dentro de la CU estén intracodificados, todas las PU dentro de la CU estén codificadas usando el modo de fusión o el modo de omisión, todos los Pus dentro de la CU estén codificados usando AMVP y un MVD para cada dirección de todas las PU que sea igual a cero. La precisión predeterminada puede, por ejemplo, ser una precisión fraccionaria o puede no ser una precisión. Por ejemplo, si un bloque está intrapredicho, entonces el bloque no tiene un vector de movimiento asociado y, por lo tanto, no tiene una precisión de vector de movimiento asociada. La primera y segunda información de precisión del vector de movimiento pueden, por ejemplo, incluir una o ambas de precisión del vector de movimiento o precisión de MVD.

Cuando la PU codificada AMVP actual se señala/deriva como con precisión de vector de movimiento de píxeles enteros, uno o más (y en algunos ejemplos, todos) candidatos MV de bloques vecinos espaciales, bloques vecinos temporales, o ambos pueden redondearse a precisión de píxeles enteros antes de podar en el proceso de generación de la lista AMVP. Cuando el MV de píxeles enteros se señala/deriva para usarse para una fusión actual, CU/PU codificada por salto, o ambos, uno o más (y en algunos ejemplos, todos) los candidatos de MV de bloques vecinos temporales espaciales, bloques vecinos temporales, o ambos, pueden redondearse a la precisión de píxeles enteros antes de podar en el proceso de generación de la lista de fusión.

Por ejemplo, el decodificador de vídeo 30 puede identificar uno o más candidatos de vector de movimiento para su inclusión en una lista de candidatos (por ejemplo, una lista de candidatos de fusión o una lista de candidatos de AMVP) para un bloque. El uno o más candidatos de vector de movimiento pueden incluir, por ejemplo, uno o más candidatos vecinos espaciales y/o uno o más candidatos vecinos temporales. Los uno o más candidatos de vector de movimiento pueden incluir al menos un candidato de vector de movimiento de precisión fraccional. En respuesta a una precisión de vector de movimiento para el bloque que es una precisión de píxel entero, el decodificador de vídeo 30 puede redondear el uno o más candidatos de vector de movimiento para determinar uno o más candidatos de vector de movimiento de precisión entera. El decodificador de vídeo 30 puede realizar una operación de poda en uno o más candidatos de vector de movimiento de precisión entera.

En un ejemplo, el indicador de precisión de vector de movimiento puede usarse (o usarse condicionalmente) como contextos CABAC de otros elementos sintácticos. Es decir, se pueden usar diferentes modelos de contexto, dependiendo del indicador de precisión del vector de movimiento, para codificar cierto elemento de sintaxis. En un ejemplo, cuando se codifica un índice candidato de AMVP para un bloque tal como PU, la (s) bandera(s) de precisión del vector de movimiento de una PU o una CU asociada o bloques espacialmente vecinos o bloques temporalmente vecinos se usa (n) como el (los) contexto(s) de codificación CABAC. De manera alternativa o adicional, en algunos ejemplos, la probabilidad inicializada de que el índice de candidatos de AMVP sea igual a 0 puede establecerse cerca de 1 cuando el indicador de precisión del vector de movimiento indica la precisión del vector de movimiento de píxeles enteros. De manera alternativa o adicional, en algunos casos, como solo en fragmentos B, o solo cuando el fragmento está a un cierto nivel temporal, o cuando el parámetro de cuantificación es mayor que un umbral predefinido, el indicador de precisión del vector de movimiento puede usarse como contextos CABAC para otros elementos sintácticos, como el índice candidato AMVP.

Se pueden combinar uno o más de estos ejemplos. Por ejemplo, en la práctica, cualquier combinación de cualquier parte del ejemplo puede usarse como nuevo ejemplo. Además, los subejemplos de los ejemplos anteriores se exponen a continuación.

Algunos ejemplos se refieren a la derivación de precisión del vector de movimiento del lado del decodificador para el contenido de la pantalla. En un ejemplo, se puede usar una plantilla en forma de L u otra forma que coincida con las muestras reconstruidas. La precisión del vector de movimiento puede basarse en la diferencia entre el resultado de coincidencia de plantilla, tal como SAD, de una posición de píxel entero y el resultado de coincidencia de su posición

de subpíxel vecina. Por ejemplo, cuando el resultado de coincidencia de la posición del píxel entero es mucho menor, se aplica la precisión del píxel entero. De lo contrario, se aplica la precisión de subpíxeles. Para definir "mucho más bajo", se puede usar un umbral. En la práctica, se puede usar un umbral fijo, un umbral adaptativo o ambos. Para un umbral adaptativo, el umbral adaptativo puede señalizarse en el flujo de bits o derivarse según otra información, tal como el tipo de bloque, o QP, señalizado en el flujo de bits. Además, también se puede definir un umbral para un caso "mucho más alto". En consecuencia, cuando el resultado de coincidencia de la posición entera menos la de la posición del subpíxel vecino es mayor que el umbral "mucho mayor", se puede usar una precisión de un cuarto de píxel. Cuando la diferencia de coincidencia está entre los umbrales de "mucho menor" y "mucho mayor", se puede usar una precisión de medio píxel. De manera alternativa o adicional, se puede usar otro procedimiento de coincidencia de plantillas, tal como la coincidencia de plantillas bidireccional basada en espejo, en su lugar en el ejemplo anterior.

En otro ejemplo, la información de precisión del vector de movimiento puede derivarse en el lado del decodificador según la propiedad de los bloques espacial o temporalmente vecinos, como el gradiente, la nitidez o si la transformada se omite para los bloques. La información de umbral puede señalizarse en el flujo de bits, derivarse del flujo de bits o ambos.

Algunos ejemplos se refieren a la señalización de indicadores. Para adaptarse de forma adaptativa a diferentes contenidos, se puede usar una combinación de diferentes procedimientos de derivación de precisión del vector de movimiento del lado del decodificador (DMPD). Para indicar qué procedimiento o procedimientos están en uso, se puede señalar un indicador en el flujo de bits. En un ejemplo, el indicador puede señalizarse a nivel de fragmento o superior para indicar explícitamente al decodificador qué procedimiento o procedimientos de DMPD se usarán. En otro ejemplo, el uso de algunos procedimientos de DMPD se señala en flujo de bits mientras que el uso de otros procedimientos de DMPD se deriva basándose en otra información, tal como tipo de fragmento y nivel temporal del fragmento, en flujo de bits.

Algunos ejemplos se refieren a la precisión del vector de movimiento adaptativo señalizado. En dicho ejemplo, la precisión del vector de movimiento puede señalizarse en el flujo de bits, como a nivel de LCU, CU o PU. Se puede usar un indicador/valor para indicar la precisión del vector de movimiento, como la precisión entera, la precisión de medio píxel, la precisión de un cuarto de píxel u otras precisiones. Cuando se señala la precisión del vector de movimiento para un bloque o una región/fragmento, todos los bloques más pequeños dentro de este bloque/región/fragmento pueden compartir la misma precisión del vector de movimiento. Además, la información MVD también puede señalizarse en la precisión señalizada. Antes de la compensación de movimiento, el MV (predicador de $MV + MVD$) se puede redondear a la precisión señalizada. El redondeo puede ser hacia infinito positivo, infinito negativo, cero o infinito (el valor negativo se redondea a infinito negativo mientras que el valor positivo se redondea a infinito positivo). De manera alternativa o adicional, el predictor MV puede redondearse primero como se mencionó anteriormente y luego formar el MV para un bloque. Después de la compensación de movimiento, el MV del bloque se guarda para la predicción MV de bloques posteriores. Al guardar el MV, el MV redondeado se puede guardar, por ejemplo, para ser usado más tarde como un candidato de fusión o candidato AMVP para un bloque decodificado posteriormente. De manera alternativa o adicional, el MV no redondeado puede guardarse en lugar del vector de movimiento redondeado, lo que puede mantener potencialmente el campo de movimiento más preciso.

En otro ejemplo, la información de precisión del vector de movimiento no se señala para el modo de omisión, el modo de fusión $2N \times 2N$ o ambos. En dicho ejemplo, la información de precisión del vector de movimiento también podría no señalizarse para una PU fusionada. De manera alternativa o adicional, las PU que se codifican en modos de codificación especiales, tales como el modo de fusión y el modo de omisión, o con particiones especiales, tales como particiones asimétricas, o con profundidad de transformada especial o con omisión de transformada, pueden mantener la precisión del vector de movimiento por defecto, tal como un cuarto de pel, incluso cuando el MV de precisión entera se señala en su nivel de CU. Alternativa o adicionalmente, otra información codificada, como el nivel temporal, QP, profundidad de CU, también se puede considerar como un modo de codificación especial o una herramienta de codificación especial.

Cuando se codifica por entropía la información de precisión del vector de movimiento con CABAC, se pueden usar contextos distintos de la información de precisión del vector de movimiento en bloques/CU espacialmente vecinos para guardar el búfer de línea, como la profundidad de la CU, la partición de la PU, el tamaño del bloque, el nivel temporal y pronto.

A continuación, se muestra el pseudocódigo, la sintaxis y la semántica de la señalización MVD de ejemplo para un bloque (por ejemplo, PU).

unidad de predicción(x0, y0, nPbW, nPbH) {	Descriptor
si(cu_skip_flag[x0][y0]) {	
si(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} si no { /*MODE INTER*/	
merge_flag[x0][y0]	ae(v)
si(merge_flag[x0][y0]) {	
si(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} si no {	
si(slice_type==B)	
inter_pred_idc[x0][y0]	ae(v)
si(inter_pred_idc[x0][y0] !=Pred_L1) {	
si(num_ref_idx_l0_active_minus1 > 0)	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0)	
mvp_l0_flag[x0][y0]	ae(v)
}	
si(inter_pred_idc[x0][y0] != Pred_L0) {	
si(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
si(mvd_l1_zero_flag && inter_pred_idc[x0][y0] ==Pred_BI) {	
MvdL1[x0][y0][0]=0	
MvdL1[x0][y0][1]=0	
}	
} si no	
mvd_coding(x0, y0, 1)	
mvp_l1_flag[x0][y0]	ae(v)
}	
si(nonZeroMvd_Exist)	
mv_precision_flag[x0][y0]	ae(v)
}	
}	
}	
mvd_coding(x0, y0, refList) {	Descriptor
abs_mvd_greater0_flag[0]	ae(v)
abs_mvd_greater0_flag[1]	ae(v)
si(abs_mvd_greater0_flag[0])	
abs_mvd_greater1_flag[0]	ae(v)

(continuación)	
unidad de predicción(x0, y0, nPbW, nPbH) {	Descriptor
si(abs_mvd_greater0_flag[1])	
abs_mvd_greater1_flag[1]	ae(v)
si(abs_mvd_greater0_flag[0]) {	
si(abs_mvd_greater1_flag[0])	
abs_mvd_minus2[0]	ae(v)
mvd_sign_flag[0]	ae(v)
}	
si(abs_mvd_greater0_flag[1]) {	
si(abs_mvd_greater1_flag[1])	
abs_mvd_minus2[1]	ae(v)
mvd_sign_flag[1]	ae(v)
}	
}	

abs_mvd_greater0_flag[0] y **abs_mvd_greater0_flag[1]** son un bin y comparten el mismo contexto.

abs_mvd_greater1_flag[0] y **abs_mvd_greater1_flag[1]** son un bin y comparten el mismo contexto.

abs_mvd_minus2[0] y **abs_mvd_minus2[1]** son un código golomb exponencial con un parámetro igual a 1.

En un ejemplo de la descripción, el uno o más elementos de sintaxis usados para indicar el MVD (por ejemplo, palabras de código que representan componentes horizontales y verticales del MVD) pueden incluir **abs_mvd_greater0_flag[x]**, **abs_mvd_greater1_flag[x]**, **abs_mvd_minus2[x]** y **mvd_sign_flag[x]**, donde [x] especifica el componente horizontal o vertical del MVD (por ejemplo, [0] para el componente horizontal y [1] para el componente vertical). El valor del elemento de sintaxis **abs_mvd_greater0_flag[x]** indica si el valor absoluto de la MVD para un componente horizontal o vertical específico es o no mayor que cero. El valor del elemento de sintaxis **abs_mvd_greater1_flag[x]** indica si el valor absoluto de la MVD para un componente horizontal o vertical específico es mayor o no que uno. Tenga en cuenta que el elemento de sintaxis **abs_mvd_greater1_flag[x]** solo se codifica si el valor del **abs_mvd_greater0_flag[x]** correspondiente indica que el valor absoluto del componente del MVD es mayor que cero. El valor de **abs_mvd_minus2[x]** indica el valor absoluto de la MVD para un componente horizontal o vertical específico. Tenga en cuenta que el elemento de sintaxis **abs_mvd_minus2[x]** solo se codifica si el valor del **abs_mvd_greater1_flag[x]** correspondiente indica que el valor absoluto del componente del MVD es mayor que uno. El valor del elemento de sintaxis **mvd_sign_flag[x]** indica la polaridad (por ejemplo, positiva o negativa) de cada componente del MVD. Lo anterior es un ejemplo de cómo se pueden codificar los MVD. Por supuesto, se pueden usar otros grupos de elementos de sintaxis para codificar un MVD.

Las técnicas de precisión de MV adaptativa (*Motion Vector Difference*, MVD) existentes solo permiten cambiar la precisión de MV/MVD entre niveles de pel enteros y pel fraccionarios. Un nivel de pel fraccional puede incluir una precisión de un cuarto de píxel (o un cuarto de muestra de luminancia) y/o una predicción de un octavo de píxel (o un octavo de muestra de luminancia). Con una precisión de un cuarto de píxel, la diferencia entre los valores de palabra de código sucesivos para un MV o un MVD es igual a un cuarto de la distancia entre píxeles (o muestras de luminancia). Del mismo modo, en la predicción de un octavo de píxel, la diferencia entre valores de palabra de código sucesivos para un MV o un MVD es igual a un octavo de la distancia entre píxeles (o muestras de luminancia). En la precisión de píxeles enteros, la diferencia entre los valores de palabras de código sucesivas para un MV o un MVD es igual a un píxel completo (o muestra de luminancia) en distancia.

Sin embargo, para el caso de secuencias de gran resolución con alto movimiento (por ejemplo, grandes cambios entre fotogramas), la señalización MV y/o MVD con precisión de pel entero puede no ser suficiente para lograr un compromiso óptimo entre la eficiencia de predicción y la sobrecarga de bits de la información MV/MVD. Es decir, los MV y los MVD pueden tener valores más grandes en secuencias de alta resolución en comparación con la secuencia de menor resolución. Los valores más grandes para MV y MVD pueden dar como resultado que las palabras de código necesiten una gran cantidad de bits para señalizar el uno o más elementos sintácticos que indican un MV y/o MVD.

Para abordar los problemas potenciales descritos anteriormente, esta descripción describe las siguientes técnicas. El codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para realizar cada una de las técnicas descritas a continuación. Las siguientes técnicas se pueden realizar individual o conjuntamente en cualquier combinación.

En un primer ejemplo de la descripción, el codificador de vídeo 20 puede configurarse para señalar, y el decodificador de vídeo 30 puede configurarse para recibir, el MVD en una unidad de múltiples muestras de luminancia (más de una muestra de luminancia entera). Es decir, en lugar de limitar la precisión de MV y/o MVD a la precisión de pel fraccional o pel entero, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para representar MV y MVD usando una precisión que es de múltiples muestras de luminancia (por ejemplo, muestras de luminancia o crominancia) a distancia. En un ejemplo, una precisión de MV/MVD de múltiples muestras puede denominarse precisión de MV/MVD de N muestras, donde N es un número entero que indica una cantidad de muestras indicadas por cada palabra de código sucesiva (por ejemplo, para componentes horizontales y verticales del MV/MVD) del uno o más elementos sintácticos que indican el MV/MVD.

Según los ejemplos de la descripción, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para determinar una precisión MV/MVD de entre tres o más precisiones MV/MVD. En este ejemplo, las tres o más precisiones MV/MVD incluyen una o más precisiones MVD de N muestras. En un ejemplo, para la señalización de MV/MVD, además de las precisiones de MV/MVD de muestra de un cuarto y muestra de número entero existentes, se puede aplicar una precisión de MVD de N muestras (por ejemplo, una muestra de luminancia y/o croma) para un bloque/CU/PU. El valor de N está predefinido y puede ser, aunque sin limitación, 2, 3 o 4. En un ejemplo, el valor de N puede señalizarse en un flujo de bits, tal como en el conjunto de parámetros de secuencia (SPS), conjunto de parámetros de imagen (PPS) o encabezado de fragmento, o cualquier otro conjunto de parámetros de nivel.

En un ejemplo de la descripción, el codificador de vídeo 20 puede configurarse para recibir un bloque de datos de vídeo a codificar usando un modo de interpredicción (por ejemplo, modo AMVP). Después de determinar un vector de movimiento y seleccionar una predicción de vector de movimiento, como se ha analizado anteriormente, el codificador de vídeo 20 puede configurarse además para determinar un MVD para el bloque de datos de vídeo. Según las técnicas de esta descripción, el codificador de vídeo 20 puede configurarse para configurarse para determinar una precisión de MVD actual, a partir de tres o más precisiones de MVD, para codificar uno o más elementos sintácticos que indican el MVD. Las tres o más precisiones de MVD incluyen una precisión de MVD de N muestras, donde N es un número entero que indica un número de muestras indicadas por cada palabra de código sucesiva del uno o más elementos sintácticos que indican el MVD, y donde N es mayor que 1. El codificador de vídeo 20 puede entonces codificar el bloque de datos de vídeo usando el MVD y el modo de interpredicción, así como codificar el uno o más elementos sintácticos que indican el MVD usando la precisión de MVD actual determinada.

De manera recíproca, el decodificador de vídeo 30 puede configurarse para recibir un bloque codificado de datos de vídeo que se codificó usando un modo de interpredicción (por ejemplo, modo AMVP), y recibir uno o más elementos sintácticos que indican un MVD asociado con el bloque codificado de datos de vídeo. Según las técnicas de esta descripción, el decodificador de vídeo 30 puede configurarse para determinar una precisión de MVD actual, a partir de tres o más precisiones de MVD, para el uno o más elementos sintácticos que indican el MVD. Las tres o más precisiones de MVD incluyen una precisión de MVD de N muestras, donde N es un número entero que indica un número de muestras indicado por cada palabra de código sucesiva del uno o más elementos sintácticos que indican el MVD, y donde N es mayor que 1. El decodificador de vídeo 30 puede configurarse además para decodificar el uno o más elementos sintácticos que indican el MVD usando la precisión de MVD actual determinada, y decodificar el bloque codificado de datos de vídeo usando el MVD decodificado. Por ejemplo, como se describió anteriormente, el decodificador de vídeo 30 puede recuperar un vector de movimiento asociado con un MVP para el bloque actual, y añadir el vector de movimiento recuperado al MVP decodificado para determinar un vector de movimiento para el bloque actual. El vector de movimiento determinado se puede usar para recuperar un bloque predictivo con el fin de realizar una interpredicción en el bloque actual.

La FIG. 8 es un diagrama conceptual que ilustra posiciones de píxeles/muestras enteras (por ejemplo, muestras de crominancia o luminancia). Las posiciones de muestra enteras que se muestran en la FIG. 8 ilustran un número de muestras a distancia de un bloque actual de datos de vídeo en la dirección horizontal, como se representa mediante un MV o un MVD. Por supuesto, las mismas técnicas pueden aplicarse para la dirección vertical. Como se muestra en la FIG. 8, dependiendo del uso de la precisión de pel entero y/o una precisión de N muestras, se puede usar una palabra de código diferente para representar la distancia indicada por el MV/MVD. Un MVD asociado con dicha palabra de código puede binarizarse en uno o más elementos sintácticos y codificarse, como se analizó anteriormente.

Para pel completo, 200 representa una distancia de un píxel/muestra desde el bloque actual en la dirección horizontal. El pel completo 201 representa una distancia de dos píxeles/muestras desde el bloque actual en la dirección horizontal. El pel completo 202 representa una distancia de tres píxeles/muestras desde el bloque actual en la dirección horizontal. El pel completo 203 representa una distancia de cuatro píxeles/muestras desde el bloque actual en la dirección horizontal. Como se muestra en la FIG. 8, la palabra de código usada para binarizar y codificar elementos de sintaxis para el componente horizontal un MVD a las diversas distancias de muestra del bloque actual es diferente dependiendo

de las precisiones de MVD usadas.

Para pel completo 200 (distancia de 1 muestra), la palabra de código para la precisión de pel MV/MVD entera es 1, mientras que la palabra de código para cada una de 2 muestras de precisión de MV/MVD, 3 muestras de precisión de MV/MVD y 4 muestras de precisión de MV/MVD es 0. Como se puede observar, en el ejemplo de una distancia de 1 muestra, 2 muestras de precisión MV/MVD, 3 muestras de precisión MV/MVD y 4 muestras de precisión MV/MVD son menos precisas (por ejemplo, porque las palabras de código se redondean hacia abajo), pero dan como resultado una palabra de código más pequeña.

Para pel completo 201 (distancia de 2 muestras), la palabra de código para la precisión de MV/MVD de pel entero es 2, la palabra de código para la precisión de MV/MVD de 2 muestras es 1, mientras que la palabra de código para cada precisión de MV/MVD de 3 muestras y la precisión de MV/MVD de 4 muestras es 0. Como se puede observar, en el ejemplo de una distancia de 2 muestras, 3 muestras de precisión MV/MVD y 4 muestras de precisión MV/MVD son menos precisas (por ejemplo, porque las palabras de código se redondean hacia abajo), pero dan como resultado una palabra de código más pequeña.

Para pel completo 202 (distancia de 3 muestras), la palabra de código para la precisión de MV/MVD de pel entero es 3, la palabra de código para la precisión de MV/MVD de 2 muestras es 1 (redondeada hacia abajo), la palabra de código para la precisión de MV/MVD de 3 muestras es 1, y la palabra de código para la precisión de MV/MVD de 4 muestras es 1 (redondeada hacia arriba). Como se puede ver, en el ejemplo de una distancia de 3 muestras, la precisión de MV/MVD de 2 muestras y la precisión de MV/MVD de 4 muestras son menos precisas (por ejemplo, porque las palabras de código se redondean hacia abajo y hacia arriba, respectivamente).

Para pel completo 203 (distancia de 4 muestras), la palabra de código para la precisión de MV/MVD de pel entero es 4, la palabra de código para la precisión de MV/MVD de 2 muestras es 2, la palabra de código para la precisión de MV/MVD de 3 muestras es 1 (redondeada hacia abajo) y la palabra de código para la precisión de MV/MVD de 4 muestras es 1. Como se puede observar, en el ejemplo de una distancia de 4 muestras, la precisión de MV/MVD de 3 muestras es menos precisa (por ejemplo, porque la palabra de código se redondea hacia abajo).

En otro ejemplo de la descripción, el codificador de vídeo 20 puede configurarse para generar un elemento de sintaxis (por ejemplo, un primer indicador) para indicar si se usa a o no una precisión de MV/MVD de un cuarto de muestra en un bloque/CU/PU. Cuando el primer indicador indica que no se usa un cuarto de MV/MVD de muestra, el codificador de vídeo 20 puede generar otro elemento sintáctico (por ejemplo, otro indicador) para indicar si se usa la precisión de MV/MVD de muestra entera o la precisión de MV/MVD de múltiples(N) muestras. El decodificador de vídeo 30 puede configurarse para recibir y analizar dichos elementos sintácticos con el fin de determinar la precisión de MV/MVD a usar para un bloque/CU/PU particular. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para decodificar los indicadores descritos anteriormente con codificación aritmética.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para determinar un contexto (por ejemplo, realizar una selección de contexto) para codificar los indicadores según el tamaño de bloque, y/o el valor de los MVD, y/o el índice de imágenes de referencia, y/o la información de precisión de MVD de bloques vecinos espaciales, y/o la diferencia de recuento de orden de imágenes (POC) de la imagen actual y la imagen de referencia, y/o cualquier otra información del bloque actual.

En otro ejemplo, el codificador de vídeo 20 se puede configurar para generar un elemento de sintaxis (por ejemplo, un primer indicador) para indicar si se usa una precisión MV/MVD específica para el bloque actual. La precisión específica de MV/MVD puede depender del valor de la precisión de MVD de los bloques vecinos espaciales. En otro ejemplo, la precisión de MVD específica se puede derivar según las estadísticas de precisión de MVD de todos los bloques codificados previamente.

En otro ejemplo, el codificador de vídeo 20 puede configurarse para generar elementos sintácticos que indican información de precisión MV/MVD para un bloque/CU/PU. En este ejemplo, los elementos de sintaxis pueden indicar que la precisión de MV/MVD se aplicará a todos los componentes (por ejemplo, componentes horizontales y verticales) de la información de MV/MVD del bloque/CU/PU. En otro ejemplo, el codificador de vídeo 20 puede configurarse para generar elementos sintácticos que indican información de precisión MVD para cada dirección de predicción (por ejemplo, predicción hacia adelante y hacia atrás), o para cada componente (por ejemplo, componentes horizontales y verticales) de la información MV/MVD del bloque/CU/PU.

En otro ejemplo, el codificador de vídeo 20 puede configurarse para generar elementos sintácticos que indican información de precisión MV/MVD para un bloque/CU/PU. En este ejemplo, los elementos de sintaxis pueden indicar que la precisión de MV/MVD se debe aplicar solo a parte de los componentes (por ejemplo, a un componente horizontal o a un componente vertical) de la información de MV de un bloque/CU/PU.

En un ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar un conjunto de información de precisión MVD por separado para cada dirección de interpredicción, índice de imagen de referencia,

diferencia POC de la imagen actual y la imagen de referencia, información de movimiento del bloque vecino o modelo de movimiento (de traslación o afín, por ejemplo).

En otro ejemplo, el codificador de vídeo 20 puede configurarse para generar elementos sintácticos que indican el conjunto de precisiones MVD permitidas que se usan para una imagen, y/o un fragmento, y/o un mosaico, y/o una secuencia. El codificador de vídeo 20 puede señalar los elementos sintácticos en el flujo de bits de vídeo codificado. En un ejemplo, el codificador de vídeo 20 puede señalar dichos elementos sintácticos en, por ejemplo, el encabezado PPS o de fragmento, para indicar que solo una parte de todas las precisiones MVD permitidas se usan para los bloques en una imagen o fragmento. Cuando se señala dicha información, la señalización de precisión MVD a nivel de bloque se cambia en consecuencia.

En otro ejemplo, la información de estas a precisiones de MVD en barbecho que se usan para una imagen, y/o un fragmento, y/o un mosaico, y/o una secuencia puede derivarse implícitamente sin señalización. La derivación del conjunto de precisiones de MVD puede depender de la resolución de secuencia, los parámetros de cuantificación, los modos de codificación y el nivel temporal de una imagen. En un ejemplo, para las imágenes que no son referidas por imágenes sucesivas (por ejemplo, el nivel temporal más alto en la estructura jerárquica), solo se permite una precisión entera o media. En un ejemplo, la precisión de MV está alineada con la precisión de MVD. En otro ejemplo, MVD puede señalizarse con una precisión que es diferente de la precisión de MV, por ejemplo, MVP se deriva con una mayor precisión.

Según los ejemplos anteriores, el codificador de vídeo 20 puede configurarse para codificar uno o más elementos sintácticos de precisión que indican la precisión de MVD actual a partir de las tres o más precisiones de MVD en uno o más de un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen o un encabezado de fragmento. En otro ejemplo, el codificador de vídeo 20 puede codificar el uno o más elementos sintácticos de precisión para uno o más de una unidad de codificación o una unidad de predicción. En un ejemplo, el uno o más elementos de sintaxis de precisión indican el valor de N para la precisión de MVD de N muestras. El codificador de vídeo 20 puede configurarse además para determinar las tres o más precisiones de MVD según una o más de una resolución de secuencia, un parámetro de cuantificación, un modo de codificación o un nivel temporal de una imagen.

Según los ejemplos anteriores, el decodificador de vídeo 30 puede configurarse para recibir uno o más elementos sintácticos de precisión que indican la precisión de MVD actual a partir de las tres o más precisiones de MVD en uno o más de un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, una unidad de codificación, una unidad de predicción o un encabezado de fragmento, y decodificar el uno o más elementos sintácticos de precisión que indican la precisión de MVD actual. En un ejemplo, el uno o más elementos de sintaxis de precisión indican el valor de N para la precisión de MVD de N muestras. El decodificador de vídeo 30 puede estar configurado además para determinar las tres o más precisiones de MVD según una o más de una resolución de secuencia, un parámetro de cuantificación, un modo de codificación o un nivel temporal de una imagen.

En otro ejemplo, cuando un MVD no se señala/recibe para un bloque (por ejemplo, en caso de modo de fusión), el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para almacenar los predictores de vector de movimiento usando la precisión de MV original (la precisión que almacena la información de MV y se usa para generar resultados de interpredicción, por ejemplo, muestra de luminancia 1/16 o muestra de luminancia 1/8 o cualquier otra precisión de MV predefinida o señalizada). Cuando un MVD se señala/recibe para un bloque (y muy probablemente la precisión de MVD también se señala/recibe), el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para redondear los candidatos de predictor de MV de bloques vecinos espaciales y/o temporales a una precisión de MV correspondiente.

En otro ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para almacenar el predictor MV con la precisión MV original. En otro ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para redondear los candidatos de predictor MV de bloques vecinos espaciales y/o temporales a otra precisión MV que es diferente a la precisión MVD del bloque actual. Por ejemplo, cuando se aplica una precisión MVD de múltiples (N) muestras a un bloque, los candidatos de predictor MV de bloques vecinos espaciales y/o temporales se redondean a una precisión de muestra de luminancia entera.

En otro ejemplo de la descripción, para un bloque que señala el MVD (por ejemplo, un bloque codificado por AMVP), si los candidatos a predictor de MV de bloques vecinos espaciales y/o temporales se redondean a la precisión de MVD correspondiente antes/después de la poda en el proceso de generación de la lista de predictores del vector de movimientos depende de si MVD es cero. En un ejemplo, si el predictor de MV debe redondearse o no depende de dónde es el predictor (por ejemplo, si el predictor es un predictor espacial o un predictor vecino). En un ejemplo, si el predictor es del bloque vecino temporal, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para redondear el vector de movimiento asociado con el predictor.

En un ejemplo, si MVD no es igual a cero para una lista de referencia, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para redondear los candidatos de predictor MV de bloques vecinos espaciales y/o temporales a una precisión MV correspondiente antes/después de la poda en el proceso de generación de lista de

predictor de vector de movimiento. De lo contrario, (por ejemplo, el MVD es igual a cero para una lista de referencia), el codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para no redondear los candidatos de MV de bloques vecinos espaciales y/o temporales en el proceso de generación de lista de predictores de vector de movimiento.

En un ejemplo, si los candidatos MV de bloques vecinos espaciales y/o temporales se redondean a la precisión MV correspondiente antes/después de la poda en el proceso de generación de la lista de predictores de vectores de movimiento depende del valor de L1ZeroMvdFlag de la imagen o bloque actual. Cuando L1ZeroMvdFlag es igual a 1 (es decir, los POC de todas las imágenes de referencia de la imagen actual son más pequeños que el POC de la imagen actual), los candidatos de MV de bloques vecinos espaciales y/o temporales se redondean a la precisión de MV correspondiente antes/después de la poda en el proceso de generación de la lista de predictores de vectores de movimiento para la lista de referencia 1; y los candidatos de MV de bloques vecinos espaciales y/o temporales no se redondean en el proceso de generación de la lista de predictores de vectores de movimiento para la lista de referencia 0. Cuando L1ZeroMvdFlag es igual a 0, los candidatos de MV de los bloques vecinos espaciales y/o temporales se redondean a la precisión de MV correspondiente antes/después de la poda en el proceso de generación de la lista de predictores de vectores de movimiento tanto para la lista de referencia 0 como para la lista de referencia 1.

La FIG. 9 es un diagrama de flujo que ilustra un procedimiento de codificación ejemplar de la descripción. Las técnicas de la FIG.9 pueden ser realizadas por uno o más componentes del codificador de vídeo 20.

En un ejemplo, el codificador de vídeo 20 puede estar configurado para recibir un bloque de datos de vídeo que se va a codificar usando un modo de interpredicción (300). El codificador de vídeo 20 puede estar configurado además para determinar un MVD para el bloque de datos de vídeo (302). El codificador de vídeo 20 puede determinar además una precisión de MVD actual, a partir de tres o más precisiones de MVD, para codificar uno o más elementos sintácticos que indican el MVD, donde las tres o más precisiones de MVD incluyen una precisión de MVD de N muestras (304). En un ejemplo, N es un número entero que indica un número de muestras indicadas por cada palabra de código sucesiva del uno o más elementos sintácticos que indican el MVD, y N es mayor que 1.

El codificador de vídeo 20 puede configurarse además para codificar el bloque de datos de vídeo usando el MVD y el modo de interpredicción (306), y codificar el uno o más elementos sintácticos que indican el MVD usando la precisión de MVD actual determinada (308).

La FIG. 10 es un diagrama de flujo que ilustra un procedimiento de decodificación ejemplar de la descripción. Las técnicas de la FIG.10 pueden ser realizadas por uno o más componentes del decodificador de vídeo 30.

En un ejemplo, el decodificador de vídeo 30 puede configurarse para recibir un bloque codificado de datos de vídeo que se codificó usando un modo de interpredicción (400), y recibir uno o más elementos sintácticos que indican un MVD asociado con el bloque codificado de datos de vídeo (402). El decodificador de vídeo 30 puede configurarse para determinar una precisión de MVD actual, a partir de tres o más precisiones de MVD, para el uno o más elementos sintácticos que indican el MVD, donde las tres o más precisiones de MVD incluyen una precisión de MVD de n muestras N (404). En un ejemplo, N es un número entero que indica un número de muestras indicadas por cada palabra de código sucesiva del uno o más elementos sintácticos que indican el MVD, y N es mayor que 1.

El decodificador de vídeo 30 puede configurarse además para decodificar el uno o más elementos sintácticos que indican el MVD usando la precisión de MVD actual determinada (406), y decodificar el bloque codificado de datos de vídeo usando el MVD decodificado (408).

En un ejemplo, el decodificador de vídeo 30 puede configurarse para recibir el uno o más elementos sintácticos de precisión cuando lo siguiente es cierto: el bloque codificado de datos de vídeo no está intracodificado, el bloque codificado de datos de vídeo no está codificado con el modo de fusión u omisión, y el bloque codificado de datos de vídeo está codificado con el modo de predicción avanzada de vectores de movimiento (*Advanced Motion Vector Prediction*, AMVP) y el MVD actual no es cero.

En un ejemplo, el decodificador de vídeo 30 puede configurarse para derivar una precisión de vector de movimiento predeterminada (por ejemplo, una precisión de muestra de un cuarto de luminancia) para su uso como la precisión MVD actual en el caso de que no se reciban uno o más elementos sintácticos de precisión.

En un ejemplo, el decodificador de vídeo 30 puede configurarse para derivar, según el modo de interpredicción, la precisión MVD actual en el caso de que no se reciban uno o más elementos sintácticos de precisión. Para derivar la precisión MVD actual, el decodificador de vídeo 30 puede configurarse además para derivar una precisión MV/MVD de muestra de luminancia de un cuarto en el caso de que el modo de interpredicción sea el modo AMVP, y derivar una precisión MV/MVD de muestra de luminancia de un octavo o una muestra de luminancia de un dieciseisavo en el caso de que el modo de interpredicción sea un modo de fusión o un modo de salto.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier

combinación de estos. Si se implementan en software, las funciones pueden almacenarse o transmitirse a través de una o más instrucciones o códigos en un medio legible por computadora 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, o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, según un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) un medio de almacenamiento tangible legible por ordenador que no sea transitorio o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder mediante uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta descripción. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo, y sin limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda usar para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión se denomina correctamente medio legible por ordenador. Por ejemplo, si se transmiten instrucciones desde un sitio web, un servidor u otra fuente remota mediante un cable coaxial, un cable de fibra óptica, un cable de par trenzado, una línea de abonado digital (*Digital Subscriber Line*, DSL) o tecnologías inalámbricas como infrarrojos, radio y microondas, el cable coaxial, el cable de fibra óptica, el cable de par trenzado, el DSL o las tecnologías inalámbricas como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por computadora y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que se refieren a medios de almacenamiento tangibles y no transitorios. Disco, como se usa en esta invención, incluye disco compacto (*Compact Disc*, CD), disco láser, disco óptico, disco versátil digital (*Digital Versatile Disc*, DVD), disquete y disco Blu-Ray, donde unos discos generalmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Dentro del alcance de los medios legibles por ordenador también deben incluirse combinaciones de lo anterior.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, como uno o más procesadores de señales digitales (*Signal Digital Processors*, DSP), microprocesadores de uso genérico, circuitos integrados específicos de aplicaciones (*Application Specific Integrated Circuits* ASIC), matrices lógicas programables de campo (*Field-Programmable Logic Arrays*, FPLA) u otros circuitos lógicos discretos o integrados equivalentes. En consecuencia, el término "procesador", como se usa en esta invención, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en esta invención. Además, en algunos ejemplos, la funcionalidad descrita en esta invención puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificación y decodificación, o incorporados en un códec combinado. Además, las técnicas se podrían implementar completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta descripción se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (*Integrated Circuit*, IC) o un conjunto de CI (por ejemplo, un conjunto de chips). Se describen varios componentes, módulos o unidades en esta descripción para enfatizar el aspecto funcional de los dispositivos configurados para realizar técnicas descritas, pero no necesariamente requieren su realización por medio de diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, varios conjuntos se pueden combinar en un conjunto de hardware de códec o proporcionar por medio de una colección de conjuntos de hardware interoperativos, que incluyen uno o más procesadores como se ha descrito anteriormente, junto con software y/o firmware adecuados.

Se han descrito varios ejemplos. El alcance de la invención está definido por las reivindicaciones adjuntas.

REIVINDICACIONES

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

5 recibir (400) un primer bloque codificado de datos de vídeo que se codificó usando un modo de interpredicción con un modelo de movimiento de traslación;
 recibir (402) uno o más primeros elementos sintácticos que indican una primera diferencia de vector de movimiento, MVD, asociada con el primer bloque codificado de datos de vídeo;
 10 determinar un primer conjunto de precisiones de MVD para la primera información de bloque codificada, donde un primer conjunto de precisiones de MVD está disponible para codificar usando el modelo de movimiento de traslación, y donde el primer conjunto de precisiones de MVD se determina a partir de tres o más precisiones de MVD que incluyen una precisión de MVD de N muestras, donde N es un número entero que indica un número de muestras indicado por cada palabra de código sucesiva del uno o más elementos de sintaxis respectivos que indican el MVD, y donde N es mayor que 1;
 15 recibir un primer elemento sintáctico de precisión de MVD indicativo de una primera precisión de MVD actual entre un primer conjunto de precisiones de MVD disponibles para el primer bloque codificado;
 determinar (404) una primera precisión de MVD actual según el primer elemento sintáctico de precisión de MVD;
 determinar un primer MVD usando la primera precisión de MVD actual determinada; y
 20 decodificar (408) el primer bloque codificado de datos de vídeo usando el primer MVD;
 recibir (400) un segundo bloque codificado de datos de vídeo que se codificó usando un modo de interpredicción con un modelo de movimiento afín;
 recibir (402) uno o más segundos elementos de sintaxis que indican un segundo MVD asociado con el segundo bloque codificado de datos de vídeo, donde un segundo conjunto de precisiones de MVD están disponibles para codificar usando el modelo de movimiento afín, donde el segundo conjunto de precisiones de MVD se determina a partir de las tres o más precisiones de MVD; recibir, separado del primer elemento de sintaxis de precisión de MVD,
 25 un segundo elemento de sintaxis de precisión de MVD indicativo de una segunda precisión MVD actual entre el segundo conjunto de precisiones MVD disponibles para el segundo bloque codificado;
 determinar (404) una segunda precisión de MVD actual basándose en el segundo elemento sintáctico de precisión de MVD; determinar un segundo MVD usando la segunda precisión de MVD actual determinada; y
 30 decodificar (408) el segundo bloque codificado de datos de vídeo usando el segundo MVD.

2. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:

35 recibir (300) un primer bloque de datos de vídeo a codificar usando un modo de interpredicción con un modelo de movimiento de traslación;
 determinar (302) una primera diferencia de vector de movimiento, MVD, para el primer bloque de datos de vídeo;
 determinar un primer conjunto de precisiones de MVD para el primer bloque, donde el primer conjunto de precisiones de MVD está disponible para codificar usando el modelo de movimiento de traslación, y donde el primer conjunto de precisiones de MVD se determina a partir de tres o más precisiones de MVD que incluyen una precisión de MVD de N muestras, donde N es un número entero que indica un número de muestras indicado por cada palabra de código sucesiva del uno o más elementos de sintaxis respectivos que indican el MVD, y donde N es mayor que 1;
 40 determinar (304) una primera precisión de MVD actual entre el primer conjunto determinado de precisiones de MVD;
 45 codificar (306) el primer bloque de datos de vídeo usando el primer MVD y el modelo de movimiento de traslación; y
 codificar (308) el uno o más primeros elementos de sintaxis que indican el primer MVD usando la primera precisión de MVD actual determinada; y
 codificar un primer elemento sintáctico de precisión de MVD indicativo de la primera precisión de MVD actual entre el primer conjunto; recibir (300) un segundo bloque de datos de vídeo a codificar usando un modo de interpredicción con un modelo de movimiento afín;
 50 determinar (302) una segunda diferencia de vector de movimiento, MVD, para el segundo bloque de datos de vídeo; determinar un segundo conjunto de precisiones MVD para el segundo bloque, donde el segundo conjunto de precisiones MVD está disponible para codificar usando el modelo de movimiento afín, y donde el segundo conjunto de precisiones MVD se determina a partir de tres o más precisiones MVD;
 55 determinar (304) una segunda precisión de MVD actual entre el segundo conjunto determinado de precisiones de MVD;
 codificar (306) el segundo bloque de datos de vídeo usando el segundo MVD y el modelo de movimiento afín;
 codificar (308) el uno o más segundos elementos de sintaxis que indican el segundo MVD usando la segunda precisión de MVD actual determinada; y
 60 codificar un segundo elemento sintáctico de precisión MVD indicativo de la segunda precisión MVD actual entre el segundo conjunto.

3. El procedimiento según la reivindicación 1 o 2, donde el modo de interpredicción es un modo de predicción de vector de movimiento avanzado.

4. El procedimiento según la reivindicación 1 o 2, donde N es una de 2, 3 o 4 muestras de luminancia.

5. El procedimiento según la reivindicación 1 o 2, donde las tres o más precisiones de MVD incluyen una precisión de MVD de muestra de luminancia de un cuarto, una precisión de MVD de muestra de luminancia entera y una o más precisiones de MVD de muestra N.

6. El procedimiento según la reivindicación 1, donde recibir el uno o más elementos sintácticos de precisión primero o segundo comprende recibir el uno o más elementos sintácticos de precisión cuando cualquiera de las siguientes condiciones es verdadera: el bloque codificado de datos de vídeo no está intracodificado, el bloque codificado de datos de vídeo no está codificado con modo de fusión u omisión, o el bloque codificado de datos de vídeo está codificado con modo de predicción avanzada de vector de movimiento AMVP, y el MVD actual no es cero.

7. El procedimiento según la reivindicación 6, comprendiendo, además: derivar una precisión de vector de movimiento predeterminada para su uso como la precisión de MVD actual en el caso de que el uno o más primeros o segundos elementos de sintaxis de precisión no se señalizan en un flujo de bits.

8. El procedimiento según la reivindicación 6, comprendiendo, además: derivar, basándose en el modo de interpredicción, la precisión MVD actual en el caso de que el uno o más elementos sintácticos de precisión no estén señalizados en el flujo de bits.

9. El procedimiento según la reivindicación 8, donde derivar la precisión MVD actual comprende derivar una precisión MVD de muestra de luminancia de un cuarto en el caso de que el modo de interpredicción sea el modo AMVP, y derivar una precisión MVD de muestra de luminancia de un octavo o de muestra de luminancia de un dieciseisavo en el caso de que el modo de interpredicción sea un modo de fusión o un modo de salto.

10. El procedimiento según la reivindicación 1, donde el uno o más elementos sintácticos de precisión primero o segundo indican el valor de N para una precisión MVD de N muestras.

11. El procedimiento según la reivindicación 1, comprendiendo, además:

recibir uno o más primeros o segundos elementos de sintaxis de precisión que indican la precisión de MVD a partir de las tres o más precisiones de MVD en uno o más de un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen o un encabezado de fragmento; y decodificar los uno o más primeros o segundos elementos de sintaxis de precisión que indican respectivamente la primera o segunda precisión de MVD actual.

12. Un aparato (30) configurado para decodificar datos de vídeo, comprendiendo:

una memoria (68) configurada para almacenar un primer bloque codificado de datos de vídeo; y uno o más procesadores en comunicación con la memoria, configurados para:

recibir el primer bloque codificado de datos de vídeo que se codificó usando un modo de interpredicción con un modelo de movimiento de traslación;

recibir uno o más primeros elementos sintácticos que indican una primera diferencia de vector de movimiento, MVD, asociada con el primer bloque codificado de datos de vídeo;

determinar un primer conjunto de precisiones de MVD para la primera información de bloque codificado, donde un primer conjunto de precisiones de MVD está disponible para codificar usando el modelo de movimiento de traslación, y donde el primer conjunto de precisiones de MVD se determina a partir de tres o más precisiones de MVD que incluyen una precisión de MVD de N muestras, donde N es un número entero que indica un número de muestras indicado por cada palabra de código sucesiva del uno o más elementos de sintaxis respectivos que indican el MVD, y donde N es mayor que 1;

recibir un primer elemento sintáctico de precisión de MVD indicativo de una primera precisión de MVD actual entre la primera pluralidad de precisiones de MVD disponibles para el primer bloque codificado;

determinar una primera precisión de MVD actual según la primera precisión de MVD;

determinar el primer MVD usando la primera precisión de MVD actual determinada; y

decodificar el primer bloque codificado de datos de vídeo usando el primer MVD; y recibir un segundo bloque codificado de datos de vídeo que se codificó usando un modo de interpredicción con un modelo de movimiento afin;

recibir uno o más segundos elementos de sintaxis que indican un segundo MVD asociado con el segundo bloque codificado de datos de vídeo, donde un segundo conjunto de precisiones de MVD están disponibles para la codificación usando el modelo de movimiento afin y donde el segundo conjunto de precisiones de MVD se determina a partir de las tres o más precisiones de MVD; recibir, separado del primer elemento de sintaxis de precisión de MVD, un segundo elemento de sintaxis de precisión de MVD indicativo de una segunda precisión de MVD actual entre el segundo conjunto de precisiones de MVD disponibles para el segundo bloque codificado;

determinar una segunda precisión de MVD actual según la segunda precisión de MVD;
determinar un segundo MVD usando la segunda precisión de MVD actual determinada; y

5 decodificar el segundo bloque codificado de datos de vídeo usando el segundo MVD.

13. Un aparato (20) configurado para codificar datos de vídeo, comprendiendo el aparato:

una memoria (38) configurada para almacenar un primer bloque de datos de vídeo; y
10 uno o más procesadores en comunicación con la memoria, el uno o más procesadores configurados para:

recibir el primer bloque de datos de vídeo a codificar usando un modo de interpredicción con un modelo de
movimiento de traslación;
determinar una primera diferencia de vector de movimiento, MVD, para el primer bloque de datos de vídeo;
determinar un primer conjunto de precisiones de MVD para el primer bloque, donde el primer conjunto de
15 precisiones de MVD está disponible para codificar usando un modelo de movimiento de traslación, y donde el
primer conjunto de precisiones de MVD se determina a partir de tres o más precisiones de MVD que incluyen
una precisión de MVD de N muestras, donde N es un número entero que indica un número de muestras indicado
por cada palabra de código sucesiva del uno o más elementos de sintaxis respectivos que indican el MVD, y
donde N es mayor que 1;
20 determinar una primera precisión de MVD actual entre el primer conjunto determinado de precisiones de MVD;
codificar el primer bloque de datos de vídeo usando el primer MVD y el modelo de movimiento de traslación;
codificar el uno o más primeros elementos sintácticos que indican el primer MVD usando la primera precisión
de MVD actual determinada; y
codificar un primer elemento sintáctico de precisión MVD indicativo de la primera precisión MVD actual entre el
25 primer conjunto; recibir un segundo bloque de datos de vídeo a codificar usando un modo de interpredicción
con un modelo de movimiento afin;
determinar una segunda diferencia de vector de movimiento, MVD, para el segundo bloque de datos de vídeo;
determinar un segundo conjunto de precisiones de MVD para el segundo bloque, donde el segundo conjunto
de precisiones de MVD está disponible para codificar usando el modelo de movimiento afin, y donde el segundo
30 conjunto de precisiones de MVD se determina a partir de tres o más precisiones de MVD;
determinar una segunda precisión de MVD actual entre el segundo conjunto determinado;
codificar el segundo bloque de datos de vídeo usando el segundo MVD y el modelo de movimiento afin;
codificar el uno o más segundos elementos de sintaxis que indican el segundo MVD usando la segunda
precisión de MVD actual determinada; y
35 codificar un segundo elemento sintáctico de precisión MVD indicativo de la segunda precisión MVD actual entre
el segundo conjunto.

14. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan,
causan uno o más procesos de un dispositivo configurado para llevar a cabo el procedimiento según cualquiera de las
40 reivindicaciones 1 a 11.

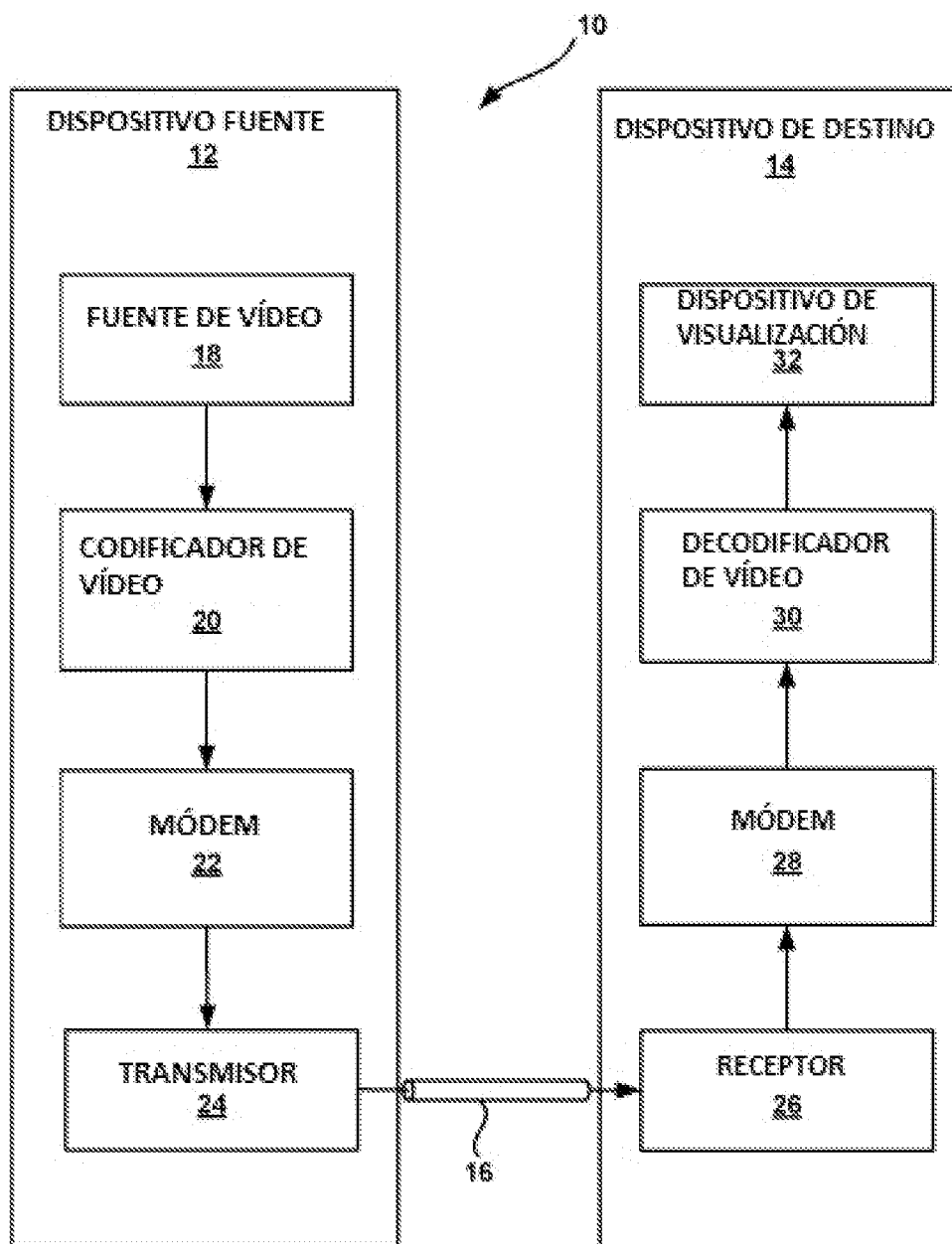


FIG. 1

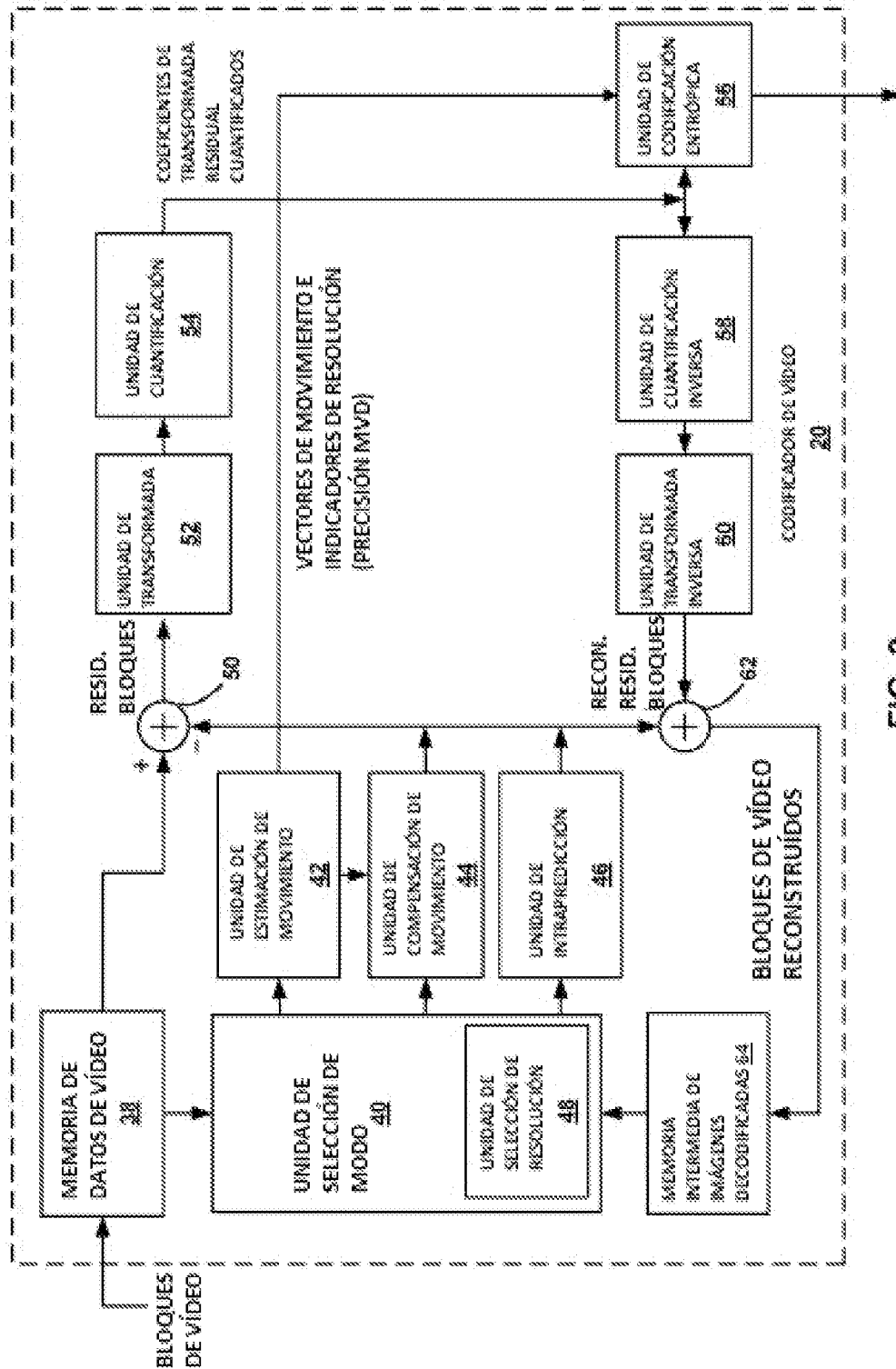


FIG. 2

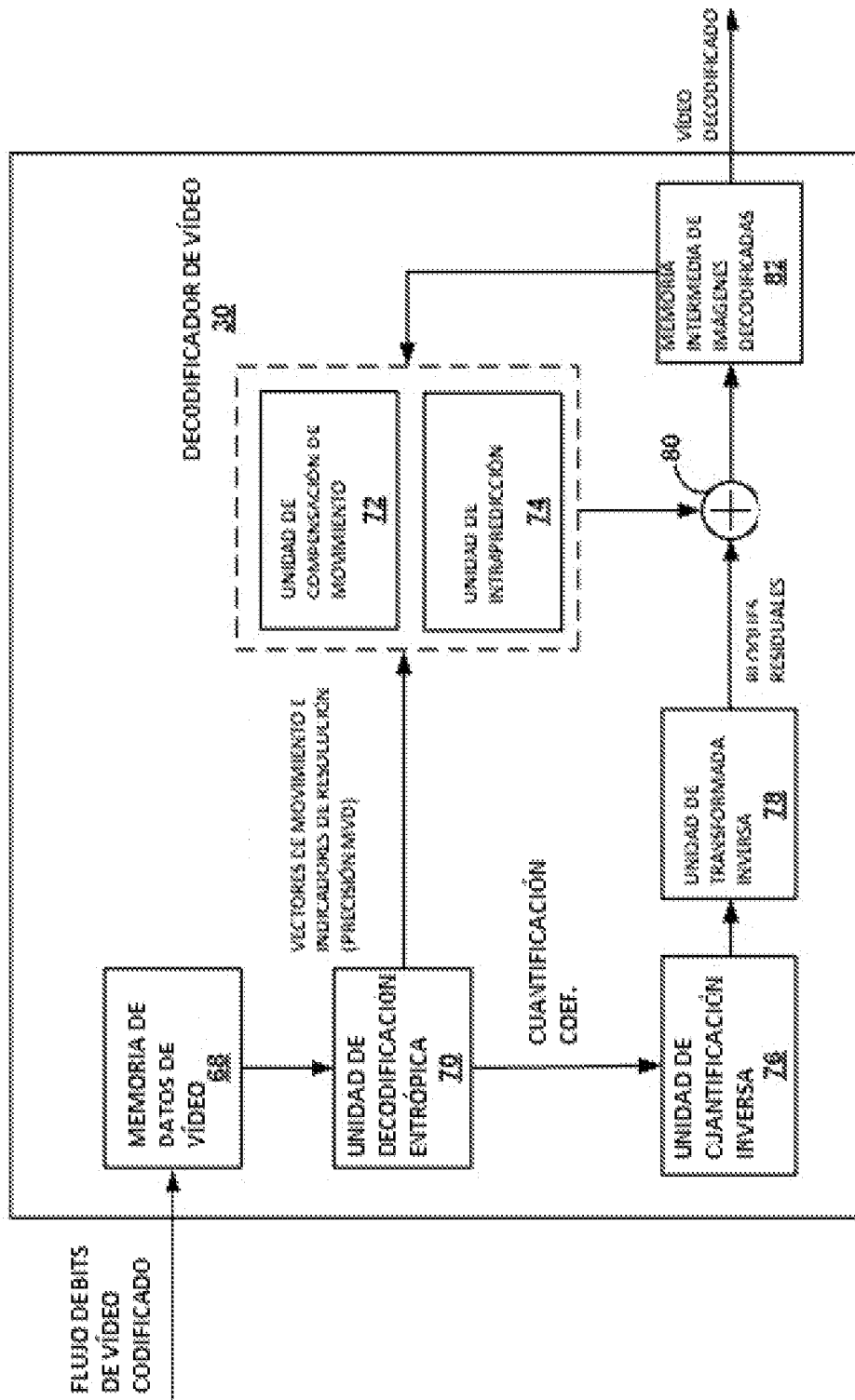
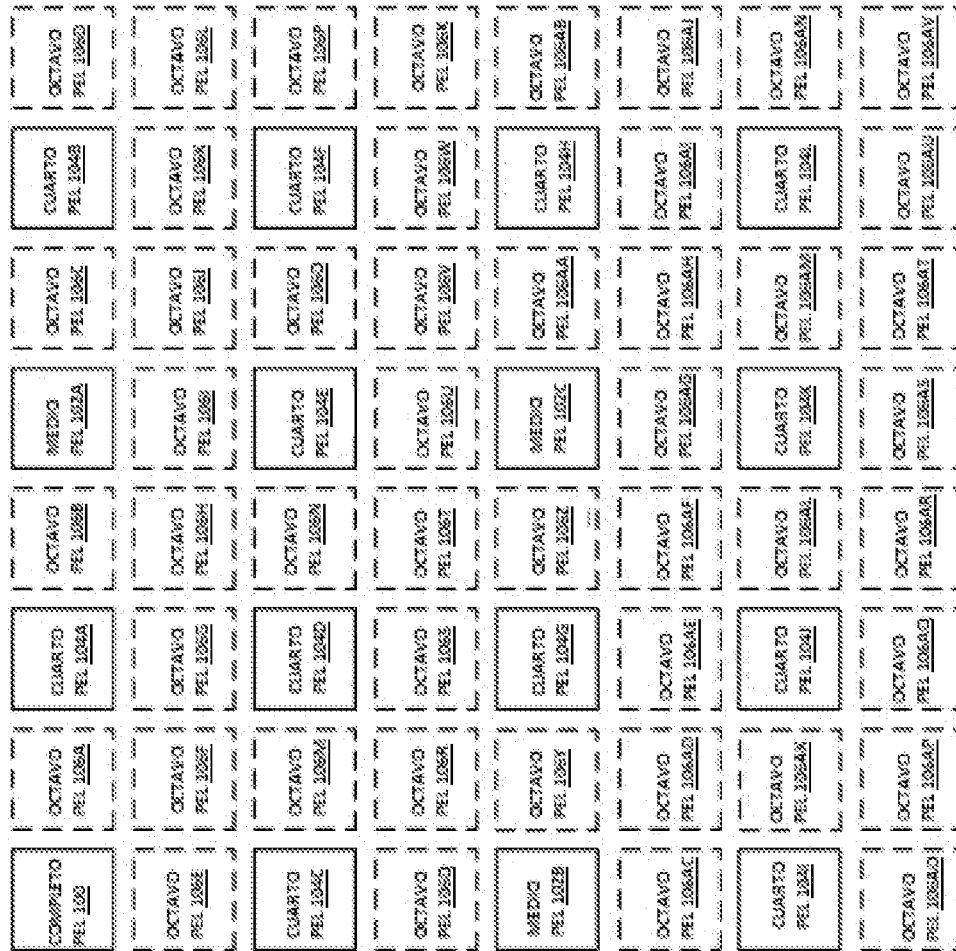
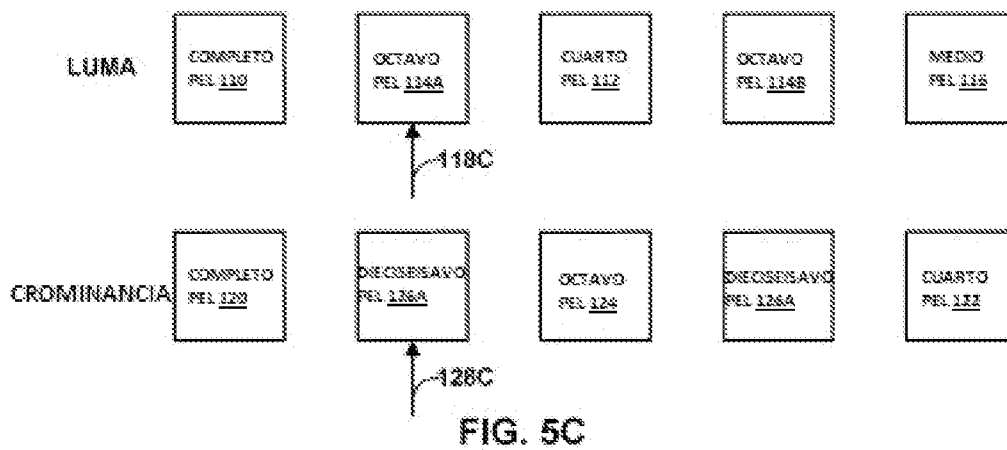
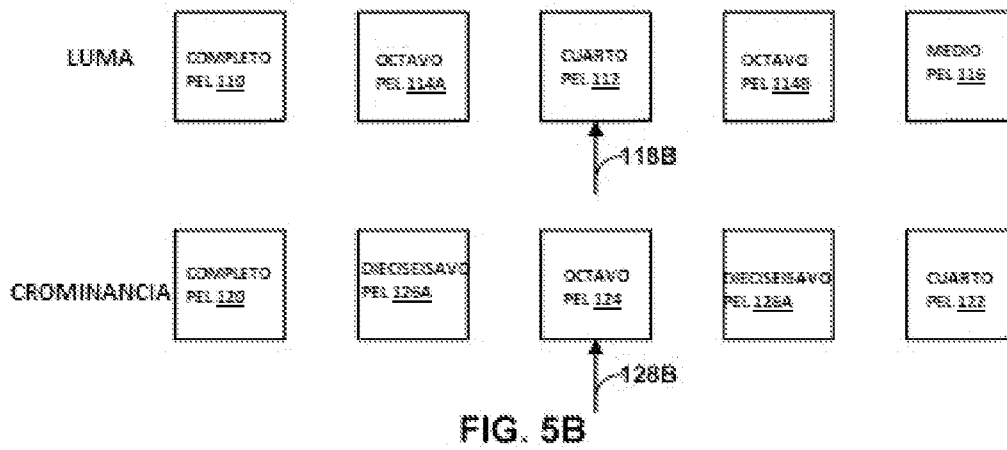
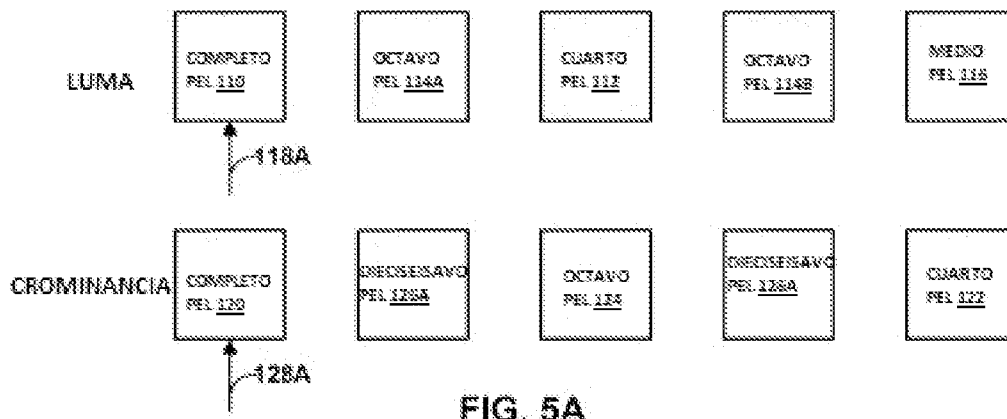


FIG. 3



401



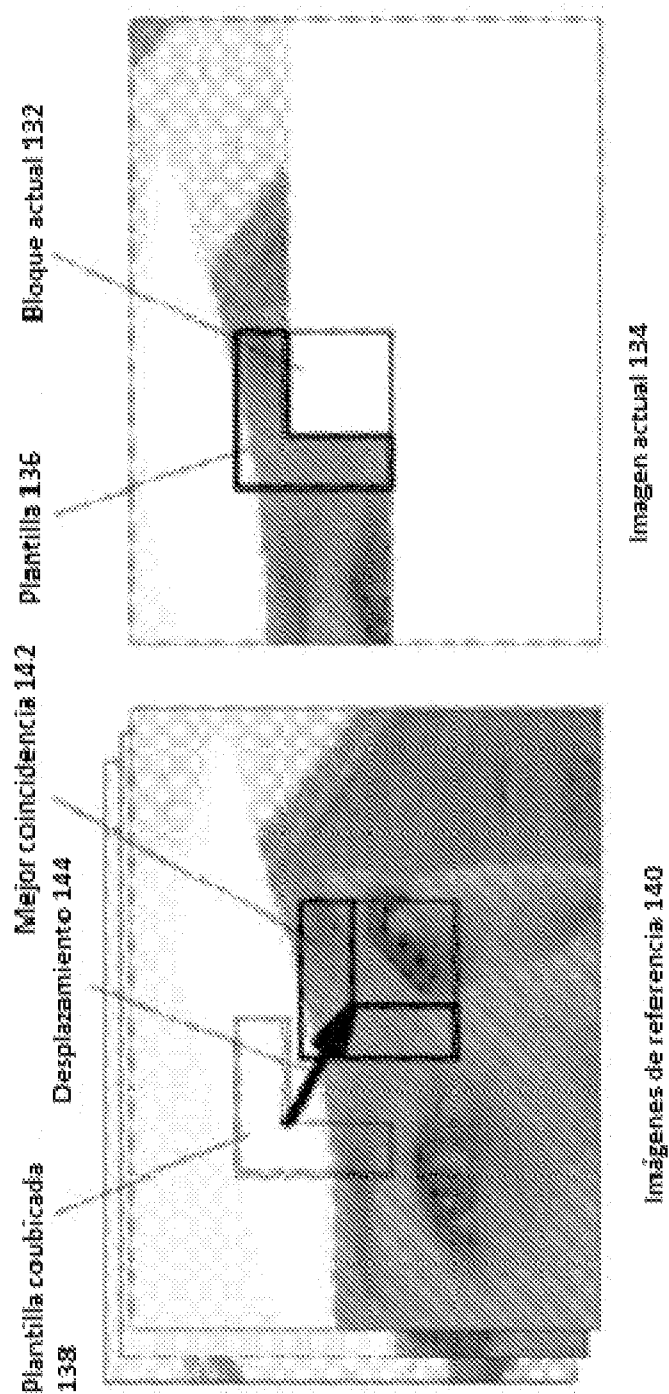


FIG. 6

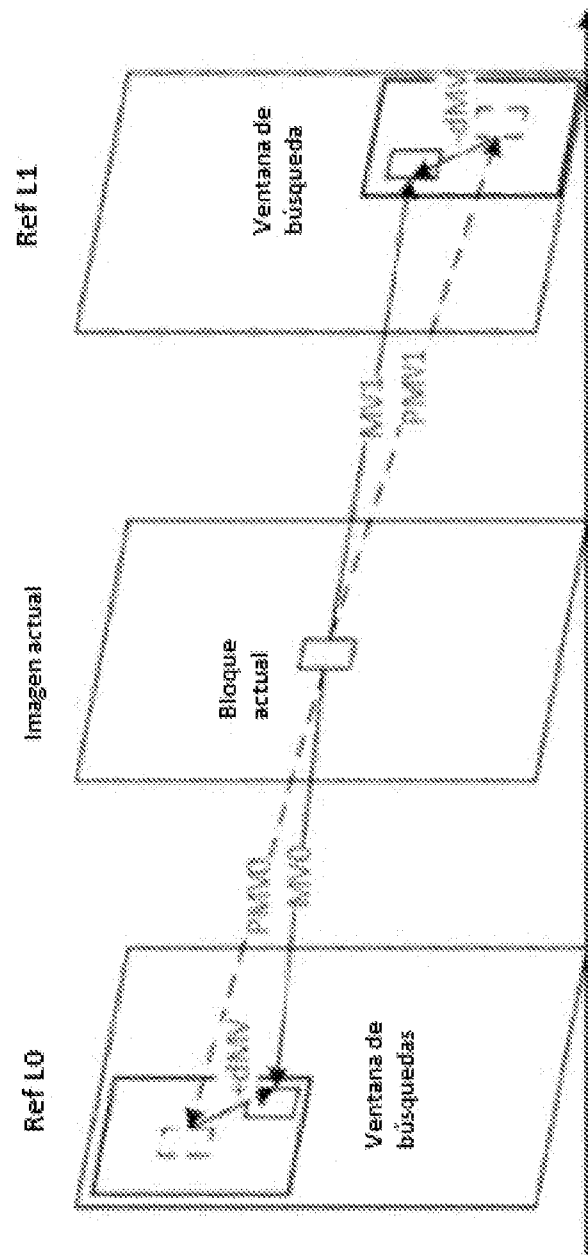


FIG. 7

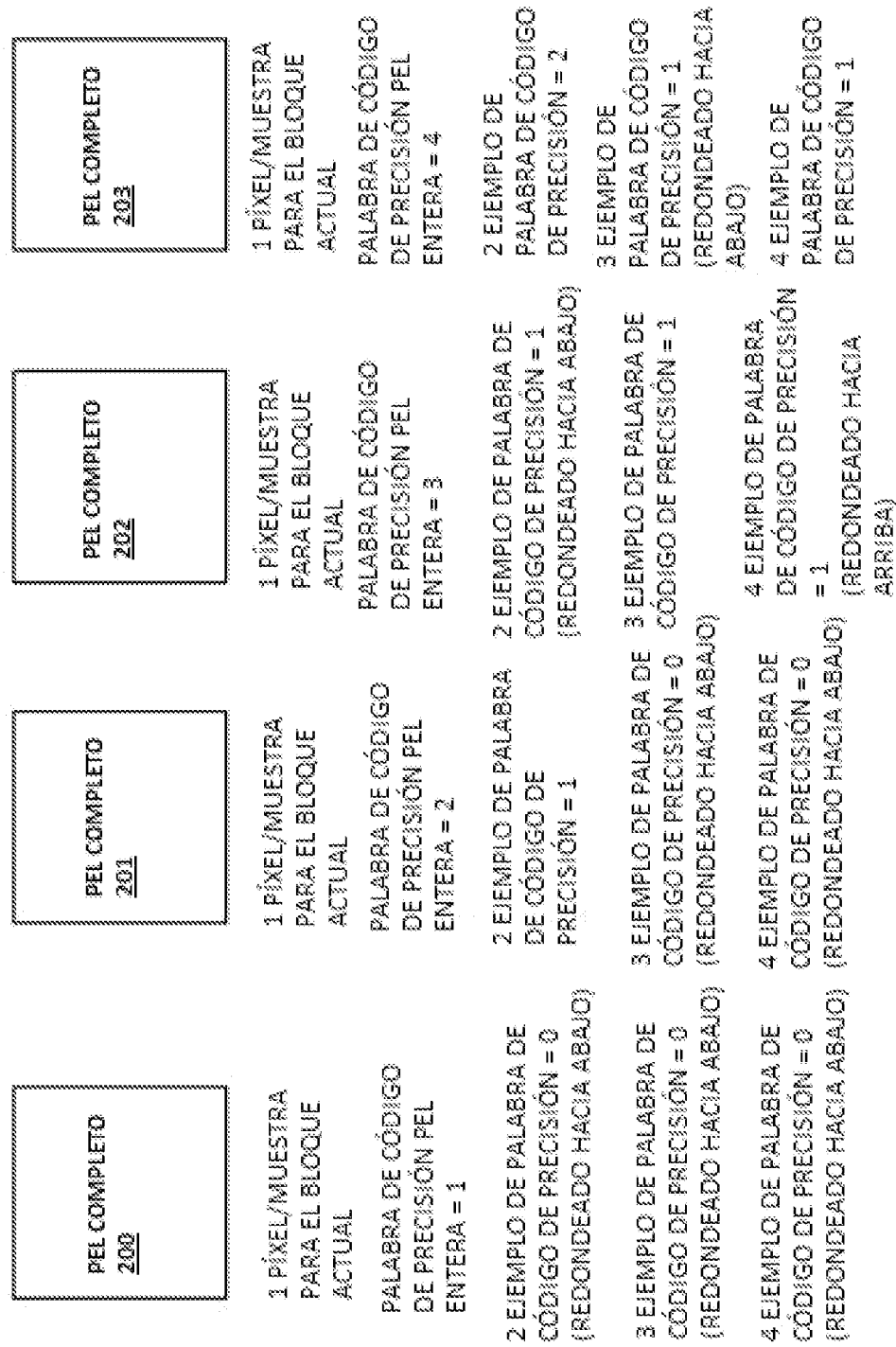


FIG. 8

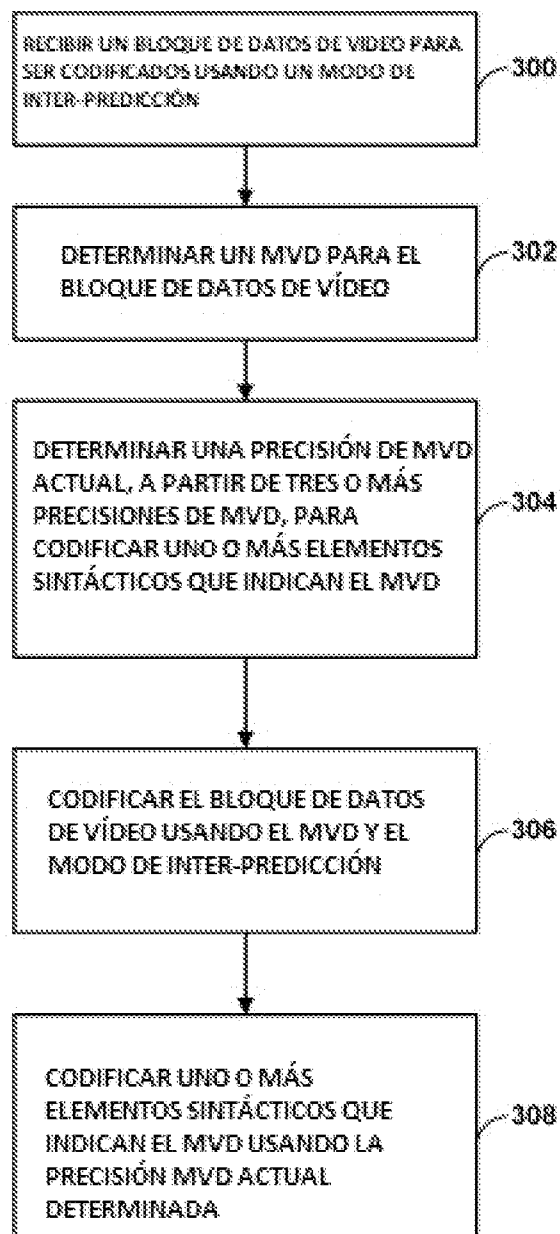


FIG. 9

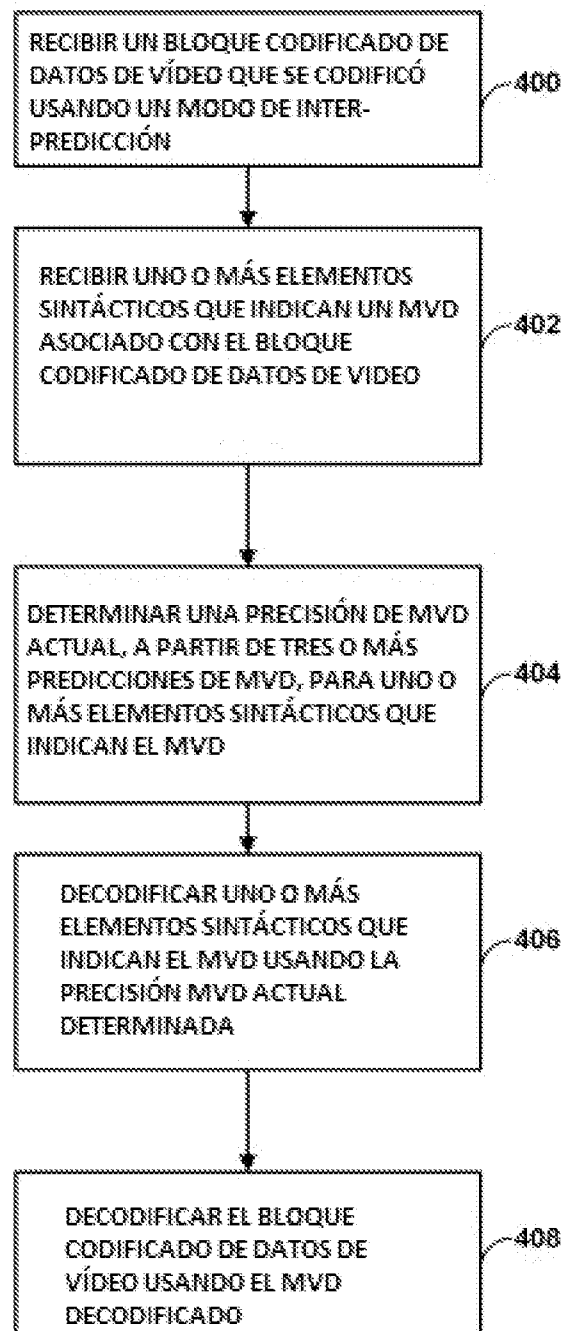


FIG. 10