

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 010 207**

51 Int. Cl.:

**H04N 19/52** (2014.01)

**H04N 19/105** (2014.01)

**H04N 19/159** (2014.01)

**H04N 19/176** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **15.08.2019 PCT/CN2019/100751**

87 Fecha y número de publicación internacional: **26.11.2020 WO20232845**

96 Fecha de presentación y número de la solicitud europea: **15.08.2019 E 19929484 (4)**

97 Fecha y número de publicación de la concesión europea: **06.11.2024 EP 3955576**

54 Título: **Método y dispositivo de interpredicción de fotogramas**

30 Prioridad:

**17.05.2019 CN 201910414914**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**01.04.2025**

73 Titular/es:

**HUAWEI TECHNOLOGIES CO., LTD. (100.00%)  
Huawei Administration Building, Bantian,  
Longgang District  
Shenzhen, Guangdong 518129, CN**

72 Inventor/es:

**CHEN, HUANBANG;  
YANG, HAITAO y  
ZHANG, LIAN**

74 Agente/Representante:

**PONS ARIÑO, Ángel**

ES 3 010 207 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**

Método y dispositivo de interpredicción de fotogramas

5 Esta solicitud reivindica prioridad a la Solicitud de Patente China N.º 201910414914.5, presentada ante la Administración Nacional de Propiedad Intelectual de China el 17 de mayo de 2019 y titulada "INTER PREDICTION METHOD AND APPARATUS".

Campo técnico

10

La presente invención se refiere al campo de codificado de vídeo y, en particular, a un método y un aparato de interpredicción para una imagen de vídeo.

Antecedentes

15

Las capacidades de vídeo digital pueden incorporarse a una amplia variedad de aparatos, incluidos televisores digitales, sistemas de transmisión digital en directo, sistemas de transmisión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, aparatos de grabación digital, reproductores de medios digitales, aparatos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite (los llamados «teléfonos inteligentes»), aparatos de videoconferencia, aparatos de transmisión de vídeo y similares. Los aparatos de vídeo digital implementan tecnologías de compresión de vídeo, por ejemplo, tecnologías de compresión de vídeo descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263 e ITU-T H.264/MPEG-4 Parte 10: codificado de vídeo avanzado (AVC), el estándar de codificado de vídeo H.265/codificado de vídeo de alta eficiencia (HEVC) 20 estándar y extensiones de dichos estándares. Los aparatos de vídeo pueden transmitir, recibir, codificar, 25 decodificar y/o almacenar información de vídeo digital de manera más eficiente mediante la implementación de dichas tecnologías de compresión de vídeo.

30

Las tecnologías de compresión de vídeo se usan para realizar predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. En el codificado de vídeo basado en bloques, un segmento de vídeo (es decir, un fotograma de vídeo o una parte de un fotograma de vídeo) se puede particionar en bloques de imagen, y el bloque de imagen también se puede denominar bloque de árbol, unidad codificadora (CU) y/o nodo de codificado. Un bloque de imagen en un segmento que se va intracodificar (I) de una imagen se codifica a través de la predicción espacial basada en una muestra de referencia en un bloque vecino en la misma imagen. Un bloque de imagen en un segmento que se va intercodificar (P o B) de una imagen se puede codificar a través de la predicción espacial basada en una muestra de referencia en un bloque vecino en la misma imagen o se puede usar predicción temporal basada en una muestra de referencia en otra imagen de referencia. La imagen puede denominarse fotograma y las imagen de referencia puede denominarse fotograma de referencia. El Codificado de Vídeo Versátil (Borrador 5), JVET-N1001-v2 revela la determinación de un vector de movimiento temporal tempMv utilizado para determinar un subbloque colocado de un subbloque de un bloque a procesar a partir de vectores de movimiento de una unidad codificadora vecina A1, que es vecino a una unidad codificadora actual a la izquierda.

45

Compendio

50

Las realizaciones de esta solicitud proporcionan un método y un aparato de interpredicción para una imagen de vídeo, y un dispositivo de codificado de vídeo correspondiente para reducir la complejidad de la implementación mientras se mejora la precisión de la predicción de información de movimiento de un bloque de imagen. La presente invención se define por las reivindicaciones.

Breve descripción de los dibujos

55

Para describir las soluciones técnicas en las realizaciones de esta solicitud o en los antecedentes más claramente, a continuación se describen brevemente los dibujos adjuntos para describir las realizaciones de esta solicitud o los antecedentes.

60

la FIG. 1A es un diagrama de bloques de un ejemplo de un sistema de codificación y decodificación de vídeo 10;

60

la FIG. 1B es un diagrama de bloques de un ejemplo de un sistema de codificado de vídeo 40;

la FIG. 2 es un diagrama de bloques de una estructura de ejemplo de un codificador 20;

65

la FIG. 3 es un diagrama de bloques de una estructura de ejemplo de un decodificador 30;

la FIG. 4 es un diagrama de bloques de un ejemplo de un dispositivo de codificado de vídeo 400;

la FIG. 5 es un diagrama de bloques de otro ejemplo de un aparato de codificación o un aparato de decodificación;

5

la FIG. 6 es un diagrama esquemático de un ejemplo de un bloque espacialmente vecino y un bloque de referencia temporal;

la FIG. 7 es un diagrama esquemático de un ejemplo de un modo de predicción AMVP;

10

la FIG. 8 es un diagrama esquemático de un ejemplo de un subbloque;

la FIG. 9 es un diagrama de flujo de un ejemplo de un método de interpredicción;

15

la FIG. 10 es un diagrama esquemático de un ejemplo de procesamiento de escala de vectores de movimiento;

la FIG. 11 es un diagrama esquemático de un ejemplo de un subbloque de un bloque a procesar y un subbloque colocado del subbloque;

20

la FIG. 12 es un diagrama de flujo de un ejemplo de otro método interpredictor; y

la FIG. 13 es un diagrama de bloques de un ejemplo de un aparato de interpredicción.

Descripción de realizaciones

25

La base para la invención reivindicada se puede encontrar en el método 3 de la realización C. Otras realizaciones deben entenderse como ejemplos de antecedentes que no son parte de la invención reivindicada. A continuación se describen las realizaciones de la presente invención con referencia a los dibujos adjuntos en las realizaciones de la presente invención. En la siguiente descripción, se hace referencia a los dibujos adjuntos, que forman parte de esta divulgación, y muestran, a modo de ilustración, aspectos específicos de las realizaciones de la presente invención o aspectos específicos en los que se pueden usar las realizaciones de la presente invención. Se debe entender que las realizaciones de la presente invención se pueden usar en otros aspectos y pueden comprender cambios estructurales o lógicos no representados en las dibujos adjuntos. Por lo tanto, la siguiente descripción detallada no debe entenderse en un sentido limitado, y el alcance de la presente invención se define mediante las reivindicaciones adjuntas. Por ejemplo, debe entenderse que el contenido divulgado con referencia a los métodos descritos también es aplicable a un dispositivo o sistema correspondiente configurado para realizar el método, y viceversa. Por ejemplo, si se describe una o más etapas del método específicas, un dispositivo correspondiente puede incluir una o más unidades tales como unidades funcionales para realizar la una o más etapas de método (por ejemplo, una unidad realiza la una o más etapas, o una pluralidad de unidades, cada una de las cuales realiza una o más de la pluralidad de etapas), incluso si tales una o más unidades no se describen o ilustran explícitamente en los dibujos adjuntos. Además, por ejemplo, si se describe un aparato específico con base en una o más unidades, tal como una unidad funcional, un método correspondiente puede incluir una etapa utilizada para realizar una o más funcionalidades de una o más unidades (por ejemplo, una etapa utilizada para realizar una o más funcionalidades de una o más unidades; o una pluralidad de etapas, cada una de los cuales se utiliza para realizar una o más funcionalidades de una o más unidades en una pluralidad de unidades), incluso si tal una o más de las etapa no se describen o ilustran explícitamente en los dibujos adjuntos. Además, se debe entender que las características de las diversas realizaciones y/o aspectos de ejemplo descritos en esta memoria descriptiva pueden combinarse entre sí, a menos que se especifique lo contrario.

50

Las soluciones técnicas en las realizaciones de la presente invención no solo se pueden aplicar a estándares de codificado de vídeo existentes (por ejemplo, estándares como H.264 y HEVC), sino también se pueden aplicar a futuros estándares de codificado de vídeo (por ejemplo, el estándar H.266). Los términos utilizados en las realizaciones de la presente invención están meramente destinados a explicar realizaciones específicas de la presente invención, pero no están destinados a limitar la presente invención. A continuación, se describen primero brevemente algunos conceptos que se pueden usar en las realizaciones de la presente invención.

55

El codificado de vídeo generalmente se refiere a procesamiento de una secuencia de imágenes, donde la secuencia de imágenes forman un vídeo o una secuencia de vídeo. En el campo de codificado de vídeo, los términos "imagen", "fotograma" o "ilustración" se pueden usar como sinónimos. El codificado de vídeo utilizado en esta memoria descriptiva indica codificación de vídeo o decodificación de vídeo. La codificación de vídeo se realiza en un lado de fuente, que incluye típicamente el procesamiento (por ejemplo, mediante compresión) de una imagen de vídeo original para reducir la cantidad de datos para representar la imagen de vídeo, para un almacenamiento y/o transmisión más eficientes. La decodificación de vídeo se realiza en un lado de destino y típicamente incluye el procesamiento inverso en comparación con un codificador para reconstruir la imagen de vídeo. El "codificado" de una imagen de vídeo en las realizaciones debe entenderse como "codificación" o

60

65

"decodificación" de una secuencia de vídeo. Una combinación de una parte de codificación y una parte de decodificación también se conoce como codificado (codificación y decodificación).

5 Una secuencia de vídeo incluye una serie de imágenes, la imagen se divide en segmentos y el segmento se divide en bloques. El codificado de vídeo se realiza por bloques. En algunos nuevos estándares de codificado de vídeo, el concepto "bloque" se amplía aún más. Por ejemplo, en el estándar H.264, hay un macrobloque (MB), y el macrobloque se puede partir aún más en una pluralidad de bloques de predicción (particiones) que se pueden usar para el codificado predictiva. En el estándar de codificado de vídeo de alta eficiencia (HEVC),  
10 una pluralidad de unidades de bloque se clasifican en función de las funciones mediante el uso de conceptos básicos como una unidad codificadora (CU), una unidad de predicción (PU) y una unidad de transformada (TU), y se describen mediante el uso de una nueva estructura basada en árboles. Por ejemplo, una CU se puede dividir en CU más pequeñas basadas en un árbol cuádruple, y la CU más pequeña se puede dividir aún más, para generar una estructura de árbol cuádruple. La CU es una unidad básica para dividir y codificar una imagen de codificación. La APU y una TU también tienen estructuras de árbol similares. La PU puede corresponder a un bloque de predicción y es una unidad básica para codificado predictivo. La CU se divide adicionalmente en una pluralidad de PU en un modo dividido. La TU puede corresponder a un bloque de transformada, y es una unidad básica para transformar una predicción residual. Sin embargo, en esencia, todas de la CU, la PU y la TU son conceptos de bloques (o bloques de imagen).

20 Por ejemplo, en HEVC, una CTU se divide en una pluralidad de CU mediante el uso de una estructura de árbol cuádruple representada como un árbol de codificado. Una decisión de codificar un área de imagen usando predicción interimagen (temporal) o intrainimagen (espacial) se toma en un nivel de CU. Cada CU se puede dividir en una, dos o cuatro PU en función de un tipo de división de PU. En una PU, se aplica un mismo proceso de predicción y la información relacionada se transmite al decodificador en función de PU. Después de obtenerse un bloque residual al aplicar el proceso de predicción en función del tipo de división de PU, la CU se puede particionar en unidades de transformada (TU) en función de otra estructura de árbol cuádruple similar al árbol de codificado de la CU. En el desarrollo reciente de tecnologías de compresión de vídeo, un fotograma de partición en árbol cuádruple y árbol binario (QTBT) se usa para particionar un bloque de codificado. En una estructura de bloques QTBT, la CU puede ser cuadrada o rectangular.

30 En esta memoria descriptiva, para facilitar la descripción y comprensión, un bloque de imagen a codificar en una imagen de codificado actual puede denominarse bloque actual. Por ejemplo, en la codificación, el bloque actual es un bloque que se está codificando, y en la decodificación, el bloque actual es un bloque que se está decodificando. Un bloque de imagen decodificado, en una imagen de referencia, utilizado para predecir el  
35 bloque actual se denomina bloque de referencia. Para ser específico, el bloque de referencia es un bloque que proporciona una señal de referencia para el bloque actual, y la señal de referencia representa un valor de píxel en el bloque de imagen. Un bloque que está en la imagen de referencia y que proporciona una señal de predicción para el bloque actual puede denominarse bloque de predicción. La señal de predicción representa un valor de píxel, un valor de muestreo o una señal de muestreo en el bloque de predicción. Por ejemplo,  
40 después de atravesarse una pluralidad de bloques de referencia, se encuentra un bloque de referencia óptimo. El bloque de referencia óptimo proporciona predicción para el bloque actual, y este bloque se conoce como un bloque de predicción.

45 En caso de codificado de vídeo sin pérdidas, una imagen de vídeo original puede reconstruirse, es decir, una imagen de vídeo reconstruida tiene la misma calidad que la imagen de vídeo original (suponiendo que no ocurre pérdida de transmisión u otra pérdida de datos durante el almacenamiento o la transmisión). En caso de codificado de vídeo con pérdida, se realiza una compresión adicional, por ejemplo, mediante cuantización, para reducir la cantidad de datos que representan las imágenes de vídeo, que no pueden reconstruirse completamente en el decodificador, es decir, la calidad de las imágenes de vídeo reconstruidas es menor o  
50 peor en comparación a la calidad de las imágenes de vídeo originales.

Varios estándares de codificado de vídeo H.261 se usan para "códecs de vídeo híbridos con pérdida" (es decir, predicción espacial y temporal en un dominio de muestra y se combina con codificado de transformada 2D para aplicar cuantización en el dominio de transformada). Cada imagen de una secuencia de vídeo usualmente  
55 se particiona en un conjunto de bloques que no se superponen y la codificación usualmente se realiza a nivel de bloque. En otras palabras, en un lado de codificador, un vídeo usualmente se procesa, es decir, se codifica, a nivel de bloque (bloque de vídeo). Por ejemplo, un bloque de predicción se genera a través de la predicción espacial (intra-imagen) y la predicción temporal (inter-imagen), el bloque de predicción se resta de un bloque actual (un bloque que actualmente se está procesando o se va a procesar) para obtener un bloque residual, y  
60 el bloque residual se transforma en el dominio de transformada y se cuantiza para reducir una cantidad de datos que se van a transmitir (comprimir). En un lado decodificador, una parte de procesamiento inversa relativa al codificador se aplica al bloque codificado o comprimido para reconstruir el bloque actual para su representación. Además, el codificador duplica un bucle de procesamiento del decodificador, de manera que el codificador y el decodificador generarán predicción idéntica (por ejemplo, intrapredicción e interpredicción)  
65 y/o reconstrucción para el procesamiento, es decir, codificado, del bloque subsiguiente.

A continuación se describe una arquitectura de sistema a la que se aplican las realizaciones de la presente invención. La FIG. 1A es un diagrama de bloques esquemático de un ejemplo de un sistema de codificación y decodificación de vídeo 10 al que se aplica una realización de la presente invención. Como se muestra en la FIG. 1A, el sistema de codificación y decodificación de vídeo 10 puede incluir un dispositivo de fuente 12 y un dispositivo de destino 14. El dispositivo de fuente 12 genera datos de vídeo codificados, y por lo tanto el dispositivo de fuente 12 puede denominarse aparato de codificación de vídeo. El dispositivo de destino 14 puede decodificar los datos de vídeo codificados generados por el dispositivo de fuente 12, y por lo tanto el dispositivo de destino 14 puede denominarse aparato de decodificación de vídeo. En diversas soluciones de implementación, el dispositivo de fuente 12, el dispositivo de destino 14, o ambos, el dispositivo de fuente 12 y el dispositivo de destino 14 pueden incluir uno o más procesadores y una memoria acoplada a uno o más procesadores. La memoria puede incluir, pero no se limita a esto, una RAM, una ROM, una EEPROM, una memoria flash o cualquier otro medio que se pueda utilizar para almacenar el código de programa deseado en una forma de instrucción o una estructura de datos accesible por ordenador, como se describe en esta memoria descriptiva. El dispositivo de fuente 12 y el dispositivo de destino 14 pueden incluir diversos aparatos, incluyendo un ordenador de escritorio, un aparato de computación móvil, un ordenador portátil (por ejemplo, un ordenador portátil), un ordenador tableta, un decodificador, un auricular telefónico tal como un llamado teléfono "inteligente", una televisión, una cámara, un aparato de visualización, un reproductor de medios digitales, una consola de videojuegos, un ordenador montada en un vehículo, un dispositivo de comunicaciones inalámbricas, o similares.

Aunque la FIG. 1A representa el dispositivo de fuente 12 y el dispositivo de destino 14 como dispositivos separados, una realización de dispositivo puede incluir el dispositivo de fuente 12 y el dispositivo de destino 14 o funcionalidades de ambos del dispositivo de fuente 12 y el dispositivo de destino 14, esto es, el dispositivo de fuente 12 o una funcionalidad correspondiente y el dispositivo de destino 14 o una funcionalidad correspondiente. En tal realización, el dispositivo de fuente 12 o la funcionalidad correspondiente y el dispositivo de destino 14 o la funcionalidad correspondiente pueden implementarse usando el mismo hardware y/o software o por hardware y/o software separados o cualquier combinación de los mismos.

Se puede implementar una conexión de comunicación entre el dispositivo de fuente 12 y el dispositivo destino 14 a través de un enlace 13, y el dispositivo destino 14 puede recibir los datos de vídeo codificados del dispositivo de fuente 12 a través del enlace 13. El enlace 13 puede incluir uno o más medios o aparatos capaces de mover los datos de vídeo codificados desde el dispositivo de fuente 12 al dispositivo de destino 14. En un ejemplo, el enlace 13 puede incluir uno o más medios de comunicación que permiten que el dispositivo de fuente 12 transmita directamente los datos de vídeo codificados al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de fuente 12 puede modular los datos de vídeo codificados según un estándar de comunicaciones (por ejemplo, un protocolo de comunicaciones inalámbricas), y puede transmitir datos de vídeo modulados al dispositivo de destino 14. El uno o más medios de comunicaciones puede incluir un medio de comunicaciones inalámbricas y/o un medio de comunicaciones por cable, por ejemplo, un espectro de radiofrecuencia (RF) o uno o más cables físicos de transmisión. El uno o más medios de comunicaciones pueden constituir una parte de una red basada en paquetes, y la red basada en paquetes es, por ejemplo, una red de área local, una red de área amplia o una red global (por ejemplo, Internet). El uno o más medios de comunicación pueden incluir un enrutador, un interruptor, una estación base u otro dispositivo que facilita la comunicación desde el dispositivo de fuente 12 al dispositivo destino 14.

El dispositivo de fuente 12 incluye un codificador 20. Opcionalmente, el dispositivo de fuente 12 puede incluir además una fuente de imagen 16, un preprocesador de imágenes 18 y una interfaz de comunicaciones 22. En una forma de implementación específica, el codificador 20, la fuente de imagen 16, el preprocesador de imágenes 18 y la interfaz de comunicaciones 22 pueden ser componentes de hardware en el dispositivo de fuente 12, o pueden ser programas de software en el dispositivo de fuente 12. Las descripciones se proporcionan por separado de la siguiente manera.

La fuente de imagen 16 puede incluir o ser cualquier tipo de dispositivo de captura de imágenes configurado para, por ejemplo, capturar una imagen del mundo real, y/o cualquier tipo de dispositivo para generar una imagen o comentario (para codificado de contenido de pantalla, algún texto en la pantalla también se considera parte de una imagen a codificar o una imagen), por ejemplo, una unidad de procesamiento de gráficos por ordenador configurada para generar una imagen de animación informática, o cualquier tipo de dispositivo configurado para obtener y/o proporcionar una imagen del mundo real o una imagen de animación informática (por ejemplo, contenido de pantalla, una imagen de realidad virtual (VR)) y/o cualquier combinación de los mismas (por ejemplo, una imagen de realidad aumentada (AR)). La fuente de imagen 16 puede ser una cámara configurada para capturar una imagen o una memoria configurada para almacenar una imagen. La fuente de imagen 16 puede incluir además cualquier tipo de interfaz (interna o externa) a través de la que se almacena una imagen previamente capturada o generada y/o se obtiene o recibe una imagen. Cuando la fuente de imagen 16 es una cámara, la fuente de imagen 16 puede ser, por ejemplo, una cámara local o una cámara integrada en el dispositivo de fuente. Cuando la fuente de imagen 16 es una memoria, la fuente de imagen 16 puede ser una memoria local o, por ejemplo, una memoria integrada en el dispositivo de fuente. Cuando la fuente de imagen 16 incluye una interfaz, la interfaz puede ser, por ejemplo, una interfaz externa para recibir

una imagen de una fuente de vídeo externa. La fuente de vídeo externa es, por ejemplo, un dispositivo de captura de imagen externo, tal como una cámara, una memoria externa o un dispositivo de generación de imagen externo. El dispositivo de generación de imágenes externo es, por ejemplo, una unidad de procesamiento de gráficos de ordenador externa, un ordenador o un servidor. La interfaz puede ser cualquier tipo de interfaz, por ejemplo, una interfaz por cable o inalámbrica, o una interfaz óptica, según cualquier protocolo de interfaz patentado o estandarizado.

Una imagen puede considerarse como una distribución bidimensional o distribución de píxeles (elementos de imagen). El píxel en la distribución también puede denominarse muestra. El número de muestras en dirección horizontal y vertical (o ejes) de la distribución o imagen define el tamaño y/o resolución de la imagen. Para la representación de un color, usualmente se utilizan tres componentes de color. Por ejemplo, la imagen puede representarse como tres distribuciones de muestras o incluir estas. Por ejemplo, en espacio de color o formato RBG, la imagen incluye una distribuciones de muestras roja, verde y azul correspondiente. Sin embargo, en el codificado de vídeo, cada píxel usualmente se representa en un formato de luma/croma o espacio de color. Por ejemplo, una imagen en un formato YUV incluye un componente de luma indicado por Y (o a veces indicado por L) y dos componentes de croma indicados por U y V. El componente de luminancia (luma) Y representa el brillo o la intensidad del nivel de gris (por ejemplo, ambos son iguales en una imagen de escala de grises), y los dos componentes de crominancia (croma) U y V representan componentes de información de croma o color. En consecuencia, la imagen en el formato YUV incluye una distribución de muestras de luma de valores de muestra de luma (Y) y dos distribuciones de muestra de croma de valores de croma (U y V). Una imagen en un formato RGB puede transformarse o convertirse en un formato YUV y viceversa. Este proceso también se conoce como conversión o transformación de color. Si una imagen es monocromática, la imagen puede incluir solo una distribución de muestras de luma. En esta realización de la presente invención, una imagen transmitida por la fuente de imagen 16 al procesador de imagen también denominarse datos de imagen sin procesar 17.

El procesador de imágenes 18 se configura para recibir los datos de imagen sin procesar 17 y realizar el preprocesamiento de los datos de imagen sin procesar 17 para obtener una imagen preprocesada 19 o datos de imagen preprocesados 19. Por ejemplo, el preprocesamiento realizado por el preprocesador de imágenes 18 puede incluir recorte, conversión de formato de color (por ejemplo, de un formato RGB a un formato YUV), corrección de color o eliminación de ruido.

El codificador 20 (o referido como codificador de vídeo 20) se configura para recibir los datos de imagen preprocesados 19, y procesar los datos de imagen preprocesados 19 en un modo de predicción relacionado (tal como un modo de predicción en las realizaciones de esta memoria descriptiva), para proporcionar datos de imagen codificada 21 (los detalles estructurales del codificador 20 se describen además a continuación con base en la FIG. 2, la FIG. 4 o la FIG. 5). En algunas realizaciones, el codificador 20 puede configurarse para realizar diversas realizaciones descritas a continuación, para implementar la aplicación del lado de codificador de un método de predicción de bloques de croma descrito en la presente invención.

La interfaz de comunicaciones 22 puede configurarse para recibir los datos de imagen codificada 21, y transmitir los datos de imagen codificada 21 al dispositivo de destino 14 o a cualquier otro dispositivo (por ejemplo, una memoria) a través del enlace 13 para almacenamiento o reconstrucción directa. El otro dispositivo puede ser cualquier dispositivo utilizado para decodificar o almacenar. La interfaz de comunicaciones 22 puede configurarse, por ejemplo, para encapsular los datos de imagen codificada 21 en un formato apropiado, por ejemplo, en un paquete de datos, para transmisión sobre el enlace 13.

El dispositivo de destino 14 incluye un decodificador 30. Opcionalmente, el dispositivo de destino 14 puede incluir además una interfaz de comunicaciones 28, un posprocesador de imagen 32 y un dispositivo de visualización 34. Las descripciones se proporcionan por separado de la siguiente manera.

La interfaz de comunicaciones 28 puede configurarse para recibir los datos de imagen codificada 21 desde el dispositivo de fuente 12 o cualquier otra fuente. La otra fuente es, por ejemplo, un dispositivo de almacenamiento. El dispositivo de almacenamiento es, por ejemplo, un dispositivo de almacenamiento de datos de imagen codificada. La interfaz de comunicaciones 28 puede configurarse para transmitir o recibir los datos de imagen codificada 21 a través del enlace 13 entre el dispositivo de fuente 12 y el dispositivo de destino 14 o a través de cualquier tipo de red. El enlace 13 es, por ejemplo, una conexión directa por cable o inalámbrica. Cualquier tipo de red es, por ejemplo, una red cableada o inalámbrica o cualquier combinación de las mismas, o cualquier tipo de red privada o pública o cualquier combinación de las mismas. La interfaz de comunicaciones 28 puede configurarse, por ejemplo, para descapsular el paquete de datos transmitido a través de la interfaz de comunicaciones 22, para obtener los datos de imagen codificada 21.

Tanto la interfaz de comunicaciones 22 como la interfaz de comunicaciones 28 pueden configurarse como interfaces de comunicaciones unidireccionales o interfaces de comunicaciones bidireccionales, y pueden configurarse, por ejemplo, para enviar y recibir mensajes para establecer una conexión, y acusar recibo e intercambiar cualquier otra información relacionada con el enlace de comunicación y/o transmisión de datos

tales como transmisión de datos de imagen codificada.

5 El decodificador 30 (o denominado decodificador 30) se configura para recibir los datos de imagen codificada 21 y proporcionar datos de imagen decodificada 31 o una imagen decodificada 31 (a continuación se describirán detalles estructurales del decodificador 30 en función de la FIG. 3, la FIG. 4 o la FIG. 5). En algunas realizaciones, el decodificador 30 puede configurarse para realizar diversas realizaciones descritas a continuación, para implementar la aplicación del lado de decodificador de un método de predicción de bloques de croma descrito en la presente invención.

10 El posprocesador de imagen 32 se configura para realizar el posprocesamiento en los datos de imagen decodificada 31 (también conocidos como datos de imagen reconstruida) para obtener datos de imagen posprocesada 33. El posprocesamiento realizado por el posprocesador de imagen 32 puede incluir conversión de formato de color (por ejemplo, de un formato YUV a un formato RGB), corrección de color, recorte, remuestreo o cualquier otro procesamiento. El posprocesador de imagen 32 puede configurarse además para  
15 transmitir los datos de imagen posprocesada 33 al dispositivo de visualización 34.

20 El dispositivo de visualización 34 se configura para recibir los datos de imagen posprocesada 33 para mostrar una imagen, por ejemplo, a un usuario o espectador. El dispositivo de visualización 34 puede ser o puede incluir cualquier tipo de pantalla para representar la imagen reconstruida, por ejemplo, una pantalla o monitor integrado o externo. Por ejemplo, las pantallas pueden incluir una pantalla de cristal líquido (LCD), una pantalla de diodos orgánicos emisores de luz (OLED), una pantalla de plasma, un proyector, un pantalla micro LED, cristal líquido sobre silicio (LCoS), procesador de luz digital (DLP) o cualquier otro tipo de pantallas.

25 Aunque la FIG. 1A representa el dispositivo de fuente 12 y el dispositivo de destino 14 como dispositivos separados, una realización de dispositivo puede incluir como alternativa ambos del dispositivo de fuente 12 y el dispositivo de destino 14 o las funcionalidades de ambos del dispositivo de fuente 12 y el dispositivo de destino 14, esto es, el dispositivo de fuente 12 o una funcionalidad correspondiente y el dispositivo de destino 14 o una funcionalidad correspondiente. En tal realización, el dispositivo de fuente 12 o la funcionalidad correspondiente y el dispositivo de destino 14 o la funcionalidad correspondiente pueden implementarse  
30 usando el mismo hardware y/o software o por hardware y/o software separados o cualquier combinación de los mismos.

35 Sobre la base de las descripciones, un experto en la técnica puede aprender definitivamente que la existencia y división (exacta) de funcionalidades de diferentes unidades o funcionalidades del dispositivo de fuente 12 y/o el dispositivo de destino 14 mostrados en la FIG. 1A, pueden variar dependiendo del dispositivo y la aplicación reales. El dispositivo de fuente 12 y el dispositivo de destino 14 pueden incluir cualquiera de una amplia gama de dispositivos, incluido cualquier tipo de dispositivo de mano o estacionario, por ejemplo, un notebook u ordenador portátil, un teléfono móvil, un teléfono inteligente, una tableta u ordenador tableta, una videocámara, un ordenador de escritorio, un decodificador, un televisor, una cámara, un dispositivo montado en vehículo,  
40 un dispositivo de visualización, un reproductor de medios digitales, una consola de videojuegos, un dispositivo de retransmisión de vídeo (tal como un servidor de servicios de contenido o un servidor de entrega de contenido), un dispositivo receptor de difusiones, un dispositivo transmisor de difusiones, y puede usar o no usar cualquier tipo de sistema operativo.

45 El codificador 20 y el decodificador 30 pueden implementarse cada uno como cualquiera de diversos circuitos adecuados, por ejemplo, uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), lógica discreta, hardware o cualquier combinación de los mismos. Si las tecnologías se implementan parcialmente usando software, un dispositivo puede almacenar instrucciones de software en un soporte de almacenamiento  
50 no transitorio legible por ordenador adecuado y puede ejecutar las instrucciones usando hardware tal como uno o más procesadores, para realizar las tecnologías de esta divulgación. Cualquiera del contenido anterior (incluyendo hardware, software, una combinación de hardware y software, y similares) puede considerarse como uno o más procesadores.

55 En algunos casos, el sistema de codificación y decodificación de vídeo 10 mostrado en la FIG. 1A es simplemente un ejemplo y las tecnologías de la presente solicitud pueden aplicarse a configuraciones de codificado de vídeo (por ejemplo, codificación de vídeo o decodificación de vídeo) que no necesariamente incluyen ninguna comunicación de datos entre un dispositivo de codificación y un dispositivo de decodificación. En otros ejemplos, pueden recuperarse datos de una memoria local, retransmitirse a través de una red o similar.  
60 Un dispositivo de codificación de vídeo puede codificar datos y almacenar datos codificados en la memoria, y/o un dispositivo de decodificación de vídeo puede recuperar datos de la memoria y decodificar los datos. En algunos ejemplos, la codificación y la decodificación la realizan dispositivos que no se comunican entre sí, sino que simplemente codifican datos para la memoria y/o recuperan datos de la memoria y decodifican los datos.

65 La FIG. 1B es un diagrama ilustrativo de un ejemplo del sistema de codificado de vídeo 40 que incluye un codificador 20 de la FIG. 2 y/o un decodificador 30 de la FIG. 3 según una realización de ejemplo. El sistema

de codificado de vídeo 40 puede implementar una combinación de diversas tecnologías en las realizaciones de la presente invención. En una implementación ilustrada, el sistema de codificado de vídeo 40 puede incluir un dispositivo de formación de imágenes 41, el codificador 20, el decodificador 30 (y/o un codificador/decodificador de vídeo implementado en un circuito lógico 47 de una unidad de procesamiento 46), una antena 42, uno o más procesadores 43, una o más memorias 44 y/o un dispositivo de visualización 45.

Como se muestra en la FIG. 1B, el dispositivo de formación de imágenes 41, la antena 42, la unidad de procesamiento 46, el circuito lógico 47, el codificador 20, el decodificador 30, el procesador 43, la memoria 44 y/o el dispositivo de visualización 45 pueden comunicarse entre sí. Como se ha descrito, aunque se ilustra el sistema de codificado de vídeo 40 con el codificador 20 y el decodificador 30, el sistema de codificado de vídeo 40 puede incluir solo el codificador 20 o solo el decodificador 30 en ejemplos diferentes.

En algunos ejemplos, la antena 42 puede configurarse para transmitir o recibir un flujo de bits codificado de datos de vídeo. Además, en algunos ejemplos, el dispositivo de visualización 45 puede configurarse para presentar los datos de vídeo. En algunos ejemplos, el circuito lógico 47 puede implementarse mediante la unidad de procesamiento 46. La unidad de procesamiento 46 puede incluir lógica de circuito integrado de aplicación específica (ASIC), una unidad de procesamiento de gráficos, un procesador de propósito general, o similares. El sistema de codificado de vídeo 40 puede incluir alternativamente un procesador opcional 43. El procesador 43 opcional puede incluir similarmente lógica de circuito integrado de aplicación específica (ASIC), una unidad de procesamiento de gráficos, un procesador de propósito general, o similares. En algunos ejemplos, el circuito lógico 47 puede implementarse utilizando hardware, por ejemplo, hardware dedicado para codificado de vídeo. El procesador 43 puede implementarse utilizando software de propósito general, un sistema operativo o similar. Además, la memoria 44 puede ser cualquier tipo de memoria, por ejemplo, una memoria volátil (por ejemplo, una memoria estática de acceso aleatorio (SRAM), una memoria dinámica de acceso aleatorio (DRAM), etc.) o una memoria no volátil (por ejemplo, memoria flash). En un ejemplo no limitativo, la memoria 44 puede implementarse mediante memoria caché. En algunos ejemplos, el circuito lógico 47 puede acceder a la memoria 44 (por ejemplo, para implementar un búfer de imágenes). En otros ejemplos, el circuito lógico 47 y/o la unidad de procesamiento 46 pueden incluir una memoria (por ejemplo, una caché) para implementar un búfer de imágenes o similar.

En algunos ejemplos, el codificador 20 implementado por el circuito lógico puede incluir un búfer de imágenes (que es implementado por, por ejemplo, la unidad de procesamiento 46 o la memoria 44) y una unidad de procesamiento de gráficos (que es implementada por, por ejemplo, la unidad de procesamiento 46). La unidad de procesamiento de gráficos puede acoplarse comunicativamente al búfer de imágenes. La unidad de procesamiento de gráficos puede incluir un codificador 20 implementado a través del circuito lógico 47 para implementar diversos módulos que se describen con referencia a la FIG. 2 y/o cualquier otro sistema o subsistema de codificador descrito en esta memoria descriptiva. El circuito lógico puede configurarse para realizar diversas operaciones descritas en esta memoria descriptiva.

En algunos ejemplos, el decodificador 30 puede implementarse por el circuito lógico 47 de manera similar, para implementar diversos módulos que se describen con referencia al decodificador 30 en la FIG. 3 y/o cualquier otro sistema o subsistema de decodificador descrito en esta memoria descriptiva. En algunos ejemplos, el decodificador 30 implementado por el circuito lógico puede incluir un búfer de imágenes (que es implementado por la unidad de procesamiento 2820 o la memoria 44) y una unidad de procesamiento de gráficos (que es implementada por, por ejemplo, la unidad de procesamiento 46). La unidad de procesamiento de gráficos puede acoplarse comunicativamente al búfer de imágenes. La unidad de procesamiento de gráficos puede incluir un decodificador 30 implementado a través del circuito lógico 47 para implementar diversos módulos que se describen con referencia a la FIG. 3 y/o cualquier otro sistema o subsistema de decodificador descrito en esta memoria descriptiva.

En algunos ejemplos, la antena 42 puede configurarse para recibir un flujo de bits codificado de datos de vídeo. Como se describe, el flujo de bits codificado puede incluir datos, un indicador, un valor de índice, datos de selección de modo, o similares que se relacionan con codificado de fotograma de vídeo y que se describe en esta memoria descriptiva, por ejemplo, datos relacionados con la partición de codificado (por ejemplo, un coeficiente de transformada o un coeficiente de transformada cuantizado, un indicador opcional (como se describe), y/o datos que definen la partición de codificado). El sistema de codificado de vídeo 40 puede incluir además el decodificador 30 que se acopla a la antena 42 y que se configura para decodificar el flujo de bits codificado. El dispositivo de visualización 45 se configura para presentar fotogramas de vídeo.

Debe entenderse que, en esta realización de la presente invención, por el ejemplo descrito con referencia al codificador 20, el decodificador 30 puede configurarse para realizar un proceso inverso. Con respecto al elementos de sintaxis de señalización, el decodificador 30 puede configurarse para recibir y analizar el elemento de sintaxis y en consecuencia decodificar datos de vídeo relacionados. En algunos ejemplos, el codificador 20 puede codificar entrópicamente el elemento de sintaxis hasta un flujo de bits de vídeo codificado. En tales ejemplos, el decodificador 30 puede analizar el elemento de sintaxis y decodificar en consecuencia los datos de vídeo relacionados.

5 Cabe señalar que un método de predicción de vectores de movimiento descrito en las realizaciones de la presente invención se utiliza principalmente en un proceso de interpredicción. Este proceso existe tanto en el codificador 20 como en el decodificador 30. El codificador 20 y el decodificador 30 en las realizaciones de la presente invención pueden ser, por ejemplo, un codificador/un decodificador que corresponde a un protocolo estándar de vídeo tal como H.263, H.264, HEVV, MPEG-2, MPEG-4, VP8 o VP9, o un protocolo estándar de vídeo de próxima generación (tal como H.266).

10 La FIG. 2 es un diagrama de bloques esquemático/conceptual de un ejemplo de un codificador 20 para implementar una realización de la presente invención. En el ejemplo de la FIG. 2, el codificador 20 incluye una unidad de cálculo residual 204, una unidad de procesamiento de transformada 206, una unidad de cuantización 208, una unidad de cuantización inversa 210, una unidad de procesamiento de transformada inversa 212, una unidad de reconstrucción 214, un búfer 216, una unidad de filtro de bucle 220, un búfer de imágenes decodificadas (DPB) 230, una unidad de procesamiento de predicción 260 y una unidad de codificación entrópica 270. La unidad de procesamiento de predicción 260 puede incluir una unidad de interpredicción 244, una unidad de intrapredicción 254 y una unidad de selección de modo 262. La unidad de interpredicción 244 puede incluir una unidad de estimación de movimiento y una unidad de compensación de movimiento (que no se muestran en la figura). El codificador 20 mostrado en la FIG. 2 también puede denominarse codificador de vídeo híbrido o un codificador de vídeo en función de un códec de vídeo híbrido.

20 Por ejemplo, la unidad de cálculo residual 204, la unidad de procesamiento de transformada 206, la unidad de cuantización 208, la unidad de procesamiento de predicción 260 y la unidad de codificación entrópica 270 forman una ruta de señal directa del codificador 20, mientras que, por ejemplo, la unidad de cuantización inversa 210, la unidad de procesamiento de transformada inversa 212, la unidad de reconstrucción 214, el búfer 216, el filtro de bucle 220, el búfer de imágenes decodificadas (DPB) 230, la unidad de procesamiento de predicción 260 forman una ruta de señal hacia atrás del codificador. La ruta de señal hacia atrás del codificador corresponde a una ruta de señal de un decodificador (consulte el decodificador 30 en la FIG. 3).

30 El codificador 20 recibe, por ejemplo, a través de una entrada 202, una imagen 201 o un bloque de imagen 203 de la imagen 201, por ejemplo, una imagen en una secuencia de imágenes que forman un vídeo o una secuencia de vídeo. El bloque de imagen 203 también puede denominarse bloque de imagen actual o bloque de imagen a codificar, y la imagen 201 puede denominarse imagen actual o imagen a codificar (en particular en codificado de vídeo, para distinguir la imagen actual de otras imágenes, por ejemplo, imágenes previamente codificadas y/o decodificadas en una misma secuencia de vídeo, es decir, la secuencia de vídeo que también incluye la imagen actual).

40 En una realización, el codificador 20 puede incluir una unidad de partición (no representada en la FIG. 2), configurada para particionar la imagen 201 en una pluralidad de bloques tales como bloques de imagen 203. La imagen 201 usualmente se particiona en una pluralidad de bloques no superpuestos. La unidad de partición puede configurarse para: usar un mismo tamaño de bloque para todas las imágenes en la secuencia de vídeo y una cuadrícula correspondiente que define el tamaño de bloque, o para cambiar un tamaño de bloque entre imágenes, subconjuntos o grupos de imágenes, y particionar cada imagen en bloques correspondientes.

45 En un ejemplo, la unidad de procesamiento de predicción 260 del codificador 20 puede configurarse para realizar cualquier combinación de tecnologías de partición descritas anteriormente.

50 Al igual que la imagen 201, el bloque de imagen 203 también se considera o puede considerarse como una distribución o matriz bidimensional de muestras con valores de muestra, aunque un tamaño del bloque de imagen 203 es más pequeño que un tamaño de la imagen 201. En otras palabras, el bloque de imagen 203 puede incluir, por ejemplo, una distribución de muestras (por ejemplo, una distribución de luma en el caso de una imagen monocromática 201), tres distribuciones de muestras (por ejemplo, una distribución de luma y dos distribuciones de croma en un caso de una imagen en color) o cualquier otra cantidad y/o tipo de distribuciones dependiendo del formato de color aplicado. Las cantidades de muestras en direcciones (o ejes) horizontal y vertical del bloque de imagen 203 define el tamaño del bloque de imagen 203.

55 El codificador 20 mostrado en la FIG. 2 se configura para codificar la imagen 201 bloque por bloque, por ejemplo, realizar codificación y predicción en cada bloque de imagen 203.

60 La unidad de cálculo residual 204 se configura para calcular un bloque residual 205 en función del bloque 203 de imágenes y de un bloque de predicción 265 (más adelante se proporcionan otros detalles sobre el bloque de predicción 265), por ejemplo, restando los valores de muestra del bloque de predicción 265 de los valores de muestra del bloque 203 de imágenes, muestra por muestra (píxel por píxel) para obtener el bloque residual 205 en un dominio de muestras.

65 La unidad de procesamiento de transformada 206 se configura para aplicar una transformada, por ejemplo, una transformada de coseno discreta (DCT) o una transformada de seno discreta (DST), a valores de muestra

del bloque residual 205 para obtener coeficientes de transformada 207 en un dominio de transformada. El coeficiente de transformada 207 también puede denominarse coeficiente residual de transformada y representa el bloque residual 205 en el dominio de transformada.

5 La unidad de procesamiento de transformada 206 puede configurarse para aplicar aproximaciones enteras de DCT/DST, tales como transformadas especificadas en HEVC/H.265. En comparación con una transformada DCT ortogonal, tal aproximación de enteros usualmente se escala mediante un factor. Para preservar una norma del bloque residual que se procesa usando transformadas directas e inversas, aplicar un factor de escala adicional es una parte del proceso de transformada. El factor de escala usualmente se elige en función de  
10 algunas restricciones. Por ejemplo, el factor de escala es una potencia de dos para una operación de desplazamiento, una profundidad de bits del coeficiente de transformada o una compensación entre la precisión y los costes de implementación. Por ejemplo, se especifica un factor de escala específico para la transformada inversa por, por ejemplo, la unidad de procesamiento de transformada inversa 212 en el lado de decodificador 30 (y una transformada inversa correspondiente por, por ejemplo, la unidad de procesamiento de transformada inversa 212 en el lado de codificador 20), y en consecuencia, se puede especificar un factor de escala correspondiente para la transformada directa por la unidad de procesamiento de transformada 206 en el lado de codificador 20.

20 La unidad de cuantización 208 se configura para cuantizar los coeficientes de transformada 207 para obtener coeficientes de transformada cuantizados 209, por ejemplo, aplicando cuantización escalar o cuantización vectorial. Los coeficientes de transformada cuantizados 209 también pueden denominarse coeficientes residuales cuantizados 209. Un proceso de cuantización puede reducir una profundidad de bits relacionada con una parte o todos los coeficientes de transformada 207. Por ejemplo, un coeficiente de transformada de n bits puede redondearse hacia abajo a un coeficiente de transformada de m bits durante la cuantización, donde  
25 n es mayor que m. Un grado de cuantización puede modificarse ajustando un parámetro de cuantización (QP). Por ejemplo, para la cuantización escalar, se pueden aplicar escalas diferentes para lograr una cuantización más fina o más gruesa. Una etapa de cuantización más pequeña corresponde a una cuantización más fina, y una etapa de cuantización más grande corresponde a una cuantización más gruesa. Una etapa de cuantización aplicable puede indicarse mediante un parámetro de cuantización (QP). Por ejemplo, el parámetro de cuantización puede ser un índice de un conjunto predefinido de etapas de cuantización aplicables. Por ejemplo,  
30 un parámetro de cuantización más pequeño puede corresponder a cuantización más fina (una etapa de cuantización más pequeña) y un parámetro de cuantización más grande puede corresponder a una cuantización más gruesa (una etapa de cuantización más grande) o viceversa. La cuantización puede incluir la división por una etapa de cuantización y la cuantización correspondiente o cuantización inversa, por ejemplo, realizada por la unidad de cuantización inversa 210, o puede incluir la multiplicación por una etapa de cuantización. En realizaciones según algunos estándares tales como HEVC, puede usarse un parámetro de cuantización para determinar la etapa de cuantización. Generalmente, la etapa de cuantización puede calcularse en función de un parámetro de cuantización usando una aproximación de punto fijo de una ecuación que incluye la división. Pueden introducirse factores de escala adicionales para la cuantización y la  
40 decuantización para restaurar la norma del bloque residual, donde la norma del bloque residual puede modificarse debido a una escala usada en la aproximación de punto fijo de la ecuación para la etapa de cuantización y el parámetro de cuantización. En una implementación de ejemplo, se pueden combinar escalas de la transformada inversa y la decuantización. Alternativamente, puede usarse una tabla de cuantización personalizada y señalarse desde el codificador al decodificador, por ejemplo, en un flujo de bits. La cuantización es una operación con pérdida, en donde una pérdida aumenta al aumentar la etapa de cuantización.

50 La unidad de cuantización inversa 210 se configura para aplicar cuantización inversa de la unidad de cuantización 208 a un coeficiente cuantizado para obtener un coeficiente decuantizado 211, por ejemplo, aplicar, sobre la base o usando una misma etapa de cuantización que la unidad de cuantización 208, la inversa de un esquema de cuantización aplicado por la unidad de cuantización 208. El coeficiente 211 decuantizado también puede denominarse coeficiente residual decuantizado 211 y corresponde al coeficiente de transformada 207, aunque el coeficiente decuantizado 211 usualmente es diferente del coeficiente de transformada debido a la pérdida provocada por cuantización.

55 La unidad de procesamiento de transformada inversa 212 se configura para aplicar una transformada inversa de la transformada aplicada por la unidad de procesamiento de transformada 206, por ejemplo, una transformada de coseno discreta inversa (DCT) o una transformada de seno discreta inversa (DST), para obtener un bloque de transformada inversa 213 en el dominio de muestras. El bloque de transformada inversa 213 también puede denominarse bloque decuantizado de transformada inversa 213 o bloque residual de transformada inversa 213.  
60

65 La unidad de reconstrucción 214 (por ejemplo, un sumador 214) se configura para añadir el bloque de transformada inversa 213 (es decir, el bloque residual reconstruido 213) al bloque de predicción 265 para obtener un bloque reconstruido 215 en el dominio de muestras, por ejemplo, añadiendo los valores de muestra del bloque residual reconstruido 213 y los valores de muestra del bloque de predicción 265.

Opcionalmente, una unidad de búfer 216 (o de forma abreviada, "búfer" 216), por ejemplo, un búfer de línea 216, se configura para almacenar en búfer o almacenar el bloque reconstruido 215 y un valor de muestra respectivos, por ejemplo para intrapredicción. En otras realizaciones, el codificador puede configurarse para usar un bloque reconstruido sin filtrar y/o un valor de muestra correspondiente que se almacenan en la unidad de búfer 216 para cualquier tipo de estimación y/o predicción, por ejemplo, intrapredicción.

Por ejemplo, en una realización, el codificador 20 puede configurarse de modo que la unidad de búfer 216 se configura para almacenar los bloques reconstruidos 215 no solo se usa para intrapredicción 254, sino también se usa para la unidad de filtro de bucle 220 (no mostrada en la FIG. 2), y/o de modo que, por ejemplo, la unidad de búfer 216 y el búfer de imágenes decodificadas 230 formen un búfer. En otras realizaciones, un bloque filtrado 221 y/o un bloque o una muestra del búfer de imágenes decodificadas 230 (el bloque o la muestra no se muestran en la FIG. 2) se usan como entrada o base para la unidad de intrapredicción 254.

La unidad de filtro de bucle 220 (o de forma abreviada "filtro de bucle" 220), se configura para filtrar el bloque reconstruido 215 para obtener el bloque filtrado 221, para suavizar la transición de píxeles o mejorar la calidad de vídeo. La unidad de filtro de bucle 220 se destina a representar uno o más filtros de bucle, por ejemplo, un filtro de des-bloqueo, un filtro de desplazamiento adaptativo de muestras (SAO) u otro filtro tal como un filtro bilateral, un filtro de bucle adaptativo (ALF), un filtro de nitidez y de suavizado o un filtro colaborativo. Aunque la unidad de filtro de bucle 220 se muestra en la FIG. 2 como un filtro en bucle, en otra configuración, la unidad de filtro de bucle 220 puede implementarse como filtro posterior al bucle. El bloque filtrado 221 también puede denominarse bloque filtrado reconstruido 221. El búfer de imágenes decodificadas 230 puede almacenar un bloque codificado reconstruido después de que la unidad de filtro de bucle 220 realice las operaciones de filtrado en el bloque codificado reconstruido.

En una realización, el codificador 20 (correspondientemente, la unidad de filtro de bucle 220) puede configurarse para tener como salida un parámetro de filtro de bucle (por ejemplo, información de desplazamiento adaptativa), por ejemplo, directamente o después de codificación entrópica por la unidad de codificación entrópica 270 o cualquier otra unidad de codificación entrópica, para que, por ejemplo, el decodificador 30 pueda recibir y aplicar el mismo parámetro de filtro de bucle para decodificar.

El búfer de imágenes decodificadas (DPB) 230 puede ser una memoria de imágenes de referencia que almacena datos de imágenes de referencia para su uso en la codificación de datos de vídeo por el codificador 20. El DPB 230 puede formarse por uno cualquiera de una variedad de dispositivos de memoria, tal como una memoria dinámica de acceso aleatorio (DRAM) (incluida una DRAM síncrona (SDRAM), una RAM magnetorresistiva (MRAM) y una RAM resistiva (RRAM)) u otro tipo de dispositivo de memoria. El DPB 230 y el búfer 216 pueden ser proporcionados por un mismo dispositivo de memoria o dispositivos de memoria separados. En un ejemplo, el búfer de imágenes decodificadas (DPB) 230 se configura para almacenar el bloque filtrado 221. El búfer de imágenes decodificadas 230 puede configurarse además para almacenar otro bloque previamente filtrado, por ejemplo, un bloque previamente reconstruido y filtrado 221, de una misma imagen actual o de diferentes imágenes, por ejemplo, una imagen previamente reconstruida, y puede proporcionar una imagen completa previamente reconstruida, es decir, decodificada (y un bloque y muestra de referencia correspondiente) y/o una imagen actual parcialmente reconstruida (y un correspondiente bloques y muestra de referencia), por ejemplo, para interpredicción. En un ejemplo, si el bloque reconstruido 215 se reconstruye sin filtrado en bucle, el búfer de imágenes decodificadas (DPB) 230 se configura para almacenar el bloque reconstruido 215.

La unidad de procesamiento de predicción 260, también denominada unidad de procesamiento de predicción de bloque 260, se configura para recibir u obtener el bloque de imagen 203 (bloque de imagen 203 actual de la imagen 201 actual) y datos de imagen reconstruida, por ejemplo, muestras de referencia de la misma imagen (actual) del búfer 216 y/o datos de imágenes de referencia 231 de una o más de imágenes previamente decodificadas del búfer de imágenes decodificadas 230; y se configura para procesar tales datos para predicción, es decir, para proporcionar el bloque de predicción 265, que puede ser un bloque de interpredicción 245 o un bloque de intrapredicción 255.

La unidad de selección de modo 262 puede configurarse para seleccionar un modo de predicción (por ejemplo, un modo de intra- o interpredicción) y/o un bloque de predicción correspondiente 245 o 255 para ser usado como bloque de predicción 265 para calcular el bloque residual 205 y para reconstruir el bloque reconstruido 215.

En una realización de la unidad de selección de modo 262 pueden configurarse para seleccionar el modo de predicción (por ejemplo, de modos de predicción soportados por la unidad de procesamiento de predicción 260), donde el modo de predicción proporciona la mejor coincidencia o un residual mínimo (el residual mínimo significa mejor compresión para transmisión o almacenamiento) o proporciona sobrecarga de señalización mínima (la sobrecarga de señalización mínima significa una mejor compresión para transmisión o almacenamiento), o que considera o equilibra ambos. La unidad de selección de modo 262 puede configurarse para determinar el modo de predicción en función de la optimización de distorsión de tasa (RDO), para ser

específicos, seleccionar un modo de predicción que proporciona una optimización de distorsión de tasa mínima o seleccionar un modo de predicción para el que la distorsión de tasa asociada satisface un criterio de selección de modo de predicción.

5 A continuación se describe en detalle el procesamiento de predicción realizado (por ejemplo, por la unidad de procesamiento de predicción 260) y la selección de modo realizada (por ejemplo, por la unidad de selección de modo 262) en un ejemplo del codificador 20.

10 Como se ha descrito anteriormente, el codificador 20 se configura para determinar o seleccionar un modo de predicción mejor u óptimo de un conjunto de modos de predicción (predeterminados). El conjunto de modos de predicción puede incluir, por ejemplo, un modo de intrapredicción y/o un modo de interpredicción.

15 Un conjunto de modos de intrapredicción puede incluir 35 modos de intrapredicción diferentes, por ejemplo, modos no direccionales tales como el modo DC (o promedio) y el modo plano, o modos direccionales, tales como se define en H.265, o puede incluir 67 modos de intrapredicción diferentes, por ejemplo, modos no direccionales como el modo DC (o promedio) y un modo plano, o modos direccionales, tales como se define en H.266 en desarrollo.

20 En una posible implementación, un conjunto de modos de interpredicción depende de las imágenes de referencia disponibles (es decir, por ejemplo, al menos una parte de las imágenes decodificadas almacenadas en el DBP 230, como se ha descrito anteriormente) y otros parámetros de interpredicción, por ejemplo, depende de si se aplica una imagen de referencia completa o solo una parte de la imagen de referencia, por ejemplo, una región de ventana de búsqueda alrededor de una región del bloque actual, para buscar un bloque de referencia de coincidencia óptima, y/o por ejemplo, depende de si se aplica la interpolación de píxeles, como  
 25 la interpolación de medio píxel y/o cuarto de píxeles. El conjunto de modos de interpredicción puede incluir, por ejemplo, un modo de predicción de vector de movimiento avanzado (AMVP) y un modo de fusión. Durante una implementación específica, el conjunto de modos de interpredicción puede incluir un modo AMVP basado en un punto de control y un modo de fusión basado en un punto de control que se mejoran en las realizaciones de la presente invención. En un ejemplo, la unidad de intrapredicción 254 puede configurarse para realizar  
 30 cualquier combinación de tecnologías de interpredicción descritas más adelante.

Además de los modos de predicción anteriores, también se puede aplicar un modo de omisión y/o un modo directo en las realizaciones de la presente invención.

35 La unidad de procesamiento de predicción 260 puede configurarse además para particionar el bloque de imagen 203 en particiones de bloque o subbloques más pequeños, por ejemplo, mediante el uso iterativo de particiones de árbol cuádruple (QT), particiones de árbol binario (BT), particiones de árbol triple (TT) o cualquier combinación de los mismos, y realizar, por ejemplo, predicción en cada una de las particiones de bloque o subbloques. La selección de modo incluye la selección de una estructura de árbol del bloque de imagen  
 40 particionado 203 y la selección de un modo de predicción aplicado para cada una de las particiones de bloque o subbloques.

45 La unidad de interpredicción 244 puede incluir una unidad de estimación de movimiento (ME) (no mostrada en la FIG. 2) y una unidad de compensación de movimiento (MC) (no mostrada en la FIG. 2). La unidad de estimación de movimiento se configura para recibir u obtener el bloque de imagen 203 (el bloque de imagen 203 actual de la imagen 201 actual) y una imagen decodificada 231, o al menos uno o más bloques previamente reconstruidos, por ejemplo, uno o más bloques reconstruidos de otras/diferentes imágenes previamente decodificadas 231, para estimación de movimiento. Por ejemplo, una secuencia de vídeo puede incluir la imagen actual y una imagen previamente decodificada 31, o en otras palabras, la imagen actual y la imagen  
 50 previamente decodificada 31 pueden ser parte o formar una secuencia de imágenes que forman una secuencia de vídeo.

55 Por ejemplo, el codificador 20 puede configurarse para seleccionar un bloque de referencia de una pluralidad de bloques de referencia de una misma imagen o diferentes imágenes en una pluralidad de otras imágenes y proporcionar, para la unidad de estimación de movimiento (no mostrada en la FIG. 2), una imagen de referencia y/o un desplazamiento (desplazamiento espacial) entre una posición (coordenadas x, y) del bloque de referencia y una posición del bloque actual como parámetro de interpredicción. Este desplazamiento también se denomina vector de movimiento (MV).

60 La unidad de compensación de movimiento se configura para obtener un parámetro de interpredicción y realizar una interpredicción en función o usando el parámetro de interpredicción para obtener un bloque de interpredicción 245. La compensación de movimiento realizada por la unidad de compensación de movimiento (no mostrada en la FIG. 2), puede incluir obtener o generar un bloque de predicción en función del vector de movimiento/bloque determinado mediante estimación de movimiento (posiblemente realizando interpolación  
 65 con precisión de subpíxel). El filtrado por interpolación puede generar muestras de píxeles adicionales a partir de muestras de píxeles conocidas. Esto aumenta potencialmente una cantidad de bloques de predicción

5 candidatos que se pueden usar para codificar un bloque de imagen. Al recibir un vector de movimiento para una PU del bloque de imagen actual, la unidad de compensación de movimiento 246 puede ubicar un bloque de predicción al que apunta el vector de movimiento en una lista de imágenes de referencia. La unidad de compensación de movimiento 246 puede generar además elementos de sintaxis asociados con un bloque y un segmento de vídeo, para decodificar un bloque de imagen del segmento de vídeo por el decodificador 30.

10 Específicamente, la unidad de interpredicción 244 puede transmitir los elementos de sintaxis a la unidad de codificación entrópica 270, donde los elementos de sintaxis incluyen el parámetro de interpredicción (tal como información de indicación de selección de un modo de interpredicción utilizado para la predicción del bloque actual después de atravesar una pluralidad de modos de interpredicción). En un posible escenario de aplicación, si solo hay un modo de interpredicción, el parámetro de interpredicción puede alternativamente no llevarse en el elemento de sintaxis. En este caso, el decodificador 30 puede realizar directamente la decodificación en un modo de predicción predeterminado. Se puede entender que la unidad de interpredicción 244 puede configurarse para realizar cualquier combinación de tecnologías de interpredicción.

15 La unidad de intrapredicción 254 se configura para obtener, por ejemplo recibir, un bloque de imagen 203 (un bloque de imagen actual) y uno o una pluralidad de bloques previamente reconstruidos, por ejemplo, bloques vecinos reconstruidos, de una misma imagen para intraestimación. Por ejemplo, el codificador 20 puede configurarse para seleccionar un modo de intrapredicción de una pluralidad de modos de intrapredicción (determinados).

20 En una realización, el codificador 20 puede configurarse para seleccionar el modo de intrapredicción según un criterio de optimización, por ejemplo, basado en residual mínimo (por ejemplo, un modo de intrapredicción que proporciona el bloque de predicción 255 que es el más similar al bloque de imagen 203 actual) o distorsión de tasa mínima.

25 La unidad de intrapredicción 254 se configura además para determinar el bloque de intrapredicción 255 en función de, por ejemplo, un parámetro de intrapredicción en el modo de intrapredicción seleccionado. En cualquier caso, después de seleccionar un modo de intrapredicción para un bloque, la unidad de intrapredicción 254 se configura además para proporcionar un parámetro de intrapredicción, es decir, información que indica el modo de intrapredicción seleccionado para el bloque, a la unidad de codificación entrópica 270. En un ejemplo, la unidad de intrapredicción 254 puede configurarse para realizar cualquier combinación de las tecnologías de intrapredicción descritas más adelante.

30 Específicamente, la unidad de intrapredicción 254 puede transmitir elementos de sintaxis a la unidad de codificación entrópica 270, donde los elementos de sintaxis incluyen el parámetro de intrapredicción (tal como información de indicación de selección de un modo de intrapredicción utilizado para la predicción del bloque actual después de atravesar una pluralidad de modos de intrapredicción). En un posible escenario de aplicación, si solo hay un modo de intrapredicción, el parámetro de intrapredicción puede alternativamente no llevarse en el elemento de sintaxis. En este caso, el decodificador 30 puede realizar directamente la decodificación en un modo de predicción predeterminado.

35 La unidad de codificación entrópica 270 se configura para aplicar (u omitir aplicar) un algoritmo o esquema de codificación entrópica (por ejemplo, un esquema de codificado de longitud variable (VLC), un esquema de VLC adaptable al contexto (CALVC), un esquema de codificación aritmética, una codificación aritmética binaria adaptable al contexto (CABAC), una codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una codificación entrópica de partición de intervalo de probabilidad (PIPE), u otra técnica o metodología de codificación entrópica) en uno o todos de los siguientes: el coeficiente cuantizado 209, el parámetro de intrapredicción, el parámetro de intrapredicción, y/o el parámetro de filtro de bucle, para obtener los datos de imagen codificada 21 que pueden salir por la salida 272, por ejemplo, en forma de un flujo de bits codificados 21. El flujo de bits codificados puede transmitirse al decodificador de vídeo 30, o archivarse para su posterior transmisión o recuperación mediante el decodificador de vídeo 30. La unidad de codificación entrópica 270 puede configurarse además para codificar entrópicamente otro elemento de sintaxis para un segmento de vídeo actual que se está codificando.

40 Se puede usar otra variante estructural del codificador de vídeo 20 para codificar un flujo de bits de vídeo. Por ejemplo, un codificador no basado en transformada 20 puede cuantizar directamente una señal residual sin la unidad de procesamiento de transformada 206 para algunos bloques o fotogramas. En otra implementación, el codificador 20 puede tener la unidad de cuantización 208 y la unidad de cuantización inversa 210 que se combinan en una sola unidad.

45 Específicamente, en esta realización de la presente invención, el codificador 20 puede configurarse para implementar un método de predicción de vector de movimiento descrito en las realizaciones siguientes.

50 Debe entenderse que otra variante estructural del codificador de vídeo 20 puede usarse para codificar un flujo de bits de vídeo. Por ejemplo, para algunos bloques de imagen o fotogramas de imagen, el codificador de vídeo

20 puede cuantizar directamente una señal residual, no se requiere procesamiento por la unidad de procesamiento de transformada 206 y, en consecuencia, tampoco se requiere procesamiento por la unidad de procesamiento de transformada inversa 212. Alternativamente, para algunos bloques de imagen o fotogramas de imagen, el codificador de vídeo 20 no genera datos residuales y, en consecuencia, no se requiere el procesamiento por la unidad de procesamiento de transformada 206, la unidad de cuantización 208, la unidad de cuantización inversa 210 y la unidad de procesamiento de transformada inversa 212. Alternativamente, el codificador de vídeo 20 puede almacenar directamente un bloque de imagen reconstruido como un bloque de referencia, y no se requiere el procesamiento por el filtro 220. Alternativamente, la unidad de cuantización 208 y la unidad de cuantización inversa 210 en el codificador de vídeo 20 pueden combinarse. El filtro de bucle 220 es opcional, y en un caso de codificado de compresión sin pérdida, la unidad de procesamiento de transformada 206, la unidad de cuantización 208, la unidad de cuantización inversa 210 y la unidad de procesamiento de transformada inversa 212 son opcionales. Debe entenderse que, según diferentes escenarios de aplicación, la unidad de interpredicción 244 y la unidad de intrapredicción 254 pueden habilitarse selectivamente.

La FIG. 3 es un diagrama de bloques esquemático/conceptual de un ejemplo de un decodificador 30 para implementar una realización de la presente invención. El decodificador de vídeo 30 se configura para recibir datos de imagen codificada (por ejemplo, un flujo de bits codificado) 21 obtenido mediante codificación por, por ejemplo, el codificador 20, para obtener una imagen decodificada 231. En un proceso de decodificación, el decodificador de vídeo 30 recibe datos de vídeo del codificador de vídeo 20, por ejemplo, un flujo de bits de vídeo codificado que representa un bloque de imagen de un segmento de vídeo codificado, y un elemento de sintaxis asociado.

En el ejemplo de la FIG. 3, el decodificador 30 incluye una unidad de decodificación entrópica 304, una unidad de cuantización inversa 310, una unidad de procesamiento de transformada inversa 312, una unidad de reconstrucción 314 (por ejemplo, un sumador 314), un búfer 316, un filtro de bucle 320, un búfer de imágenes decodificadas 330 y una unidad de procesamiento de predicción 360. La unidad de procesamiento de predicción 360 puede incluir una unidad de interpredicción 344, una unidad de intrapredicción 354 y una unidad de selección de modo 362. En algunos ejemplos, el decodificador de vídeo 30 puede realizar una etapa de decodificación generalmente recíproca a la etapa de codificación descrita con referencia al codificador de vídeo 20 en la FIG. 2.

La unidad de decodificación entrópica 304 se configura para realizar decodificación entrópica de los datos de imagen codificada 21 para obtener, por ejemplo, un coeficiente cuantizados 309 y/o un parámetro de codificado decodificado (no mostrado en la FIG. 3), por ejemplo, uno cualquiera o todos los parámetros de interpredicción, un parámetro de intrapredicción, un parámetro de filtro de bucle, y/o otro elementos de sintaxis (que se decodifican). La unidad de decodificación entrópica 304 se configura además para reenviar el parámetro de interpredicción, el parámetro de intrapredicción y/o el otro elemento de sintaxis a la unidad de procesamiento de predicción 360. El decodificador de vídeo 30 puede recibir elementos de sintaxis en un nivel de segmento de vídeo y/o un nivel de bloque de vídeo.

La unidad de cuantización inversa 310 puede tener la misma función que la unidad de cuantización inversa 110. La unidad de procesamiento de transformada inversa 312 puede tener una misma función que la unidad de procesamiento de transformada inversa 212. La unidad de reconstrucción 314 puede tener una misma función que la unidad de reconstrucción 214. El búfer 316 puede tener la misma función que el búfer 216. El filtro de bucle 320 puede tener la misma función que el filtro de bucle 220. El búfer de imágenes decodificadas 330 puede tener la misma función que el búfer de imágenes decodificadas 230.

La unidad de procesamiento de predicción 360 puede incluir la unidad de interpredicción 344 y la unidad de intrapredicción 354. Una función de la unidad de interpredicción 344 puede ser similar a una función de la unidad de interpredicción 244, y una función de la unidad de intrapredicción 354 puede ser similar a una función de la unidad de intrapredicción 254. La unidad de procesamiento de predicción 360 usualmente se configura para realizar la predicción de bloque y/u obtener el bloque de predicción 365 a partir de los datos codificados 21 y recibir u obtener (explícita o implícitamente) un parámetro relacionado con predicción y/o información sobre el modo de predicción seleccionado, por ejemplo, de la unidad de decodificación entrópica 304.

Cuando un segmento de vídeo se codifica como un segmento intracodificado (I), la unidad de intrapredicción 354 de la unidad de procesamiento de predicción 360 se configura para generar el bloque de predicción 365 para un bloque de imagen del segmento de vídeo actual en función de un modo de intrapredicción señalado y datos que son de un bloque previamente decodificado del fotograma o imagen actual. Cuando un fotograma de vídeo se codifica como segmento intercodificado (B o P), la unidad de interpredicción 344 (por ejemplo, una unidad de compensación de movimiento) de la unidad de procesamiento de predicción 360 se configura para generar el bloque de predicción 365 para un bloque de vídeo del segmento de vídeo actual en función de un vector de movimiento y otro elemento de sintaxis que se recibe de la unidad de decodificación entrópica 304. Para la interpredicción, el bloque de predicción pueden generarse a partir de una de imagen de referencia en una lista de imágenes de referencia. El decodificador de vídeo 30 puede construir listas de fotogramas de referencia, una lista 0 y una lista 1, usando una tecnología de construcción predeterminada basada en

imágenes de referencia almacenadas en la DPB 330.

La unidad de procesamiento de predicción 360 se configura para determinar información de predicción para el bloque de vídeo del segmento de vídeo actual al analizar el vector de movimiento y otro elemento de sintaxis, y usar la información de predicción para generar un bloque de predicción para el bloque de vídeo actual que se está decodificando. En un ejemplo de la presente invención, la unidad de procesamiento de predicción 360 determina, usando algunos elementos de sintaxis recibidos, un modo de predicción (por ejemplo, intra- o interpredicción) para codificar el bloque de vídeo del segmento de vídeo, un tipo de segmento de interpredicción (por ejemplo, segmento B, segmento P o segmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el segmento, un vector de movimiento para cada bloque de vídeo intercodificado del segmento, un estado de interpredicción para cada bloque de vídeo intercodificado del segmento, y otra información para decodificar el bloque de vídeo en el segmento de vídeo actual. En otro ejemplo de esta divulgación, los elementos de sintaxis recibidos por el decodificador de vídeo 30 desde un flujo de bits incluyen elementos de sintaxis en uno o más de un conjunto de parámetros adaptativos (APS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un encabezado de segmento.

La unidad de cuantización inversa 310 se configura para realizar cuantización inversa ( es decir, decuantización) en un coeficiente de transformada cuantizado que se proporcionan en el flujo de bits y se decodifica por la unidad de decodificación entrópica 304. Un proceso de cuantización inversa puede incluir: usar un parámetro de cuantización calculado por el codificador de vídeo 20 para cada bloque de vídeo en el segmento de vídeo, para determinar un grado de cuantización que debe aplicarse y, asimismo, un grado de cuantización inversa que debe aplicarse.

La unidad de procesamiento de transformada inversa 312 se configura para aplicar una transformada inversa (por ejemplo, una DCT inversa, una transformada entera inversa o un proceso de transformada inversa conceptualmente similar) a un coeficiente de transformada, para generar un bloque residual en el dominio de píxeles.

La unidad de reconstrucción 314 (por ejemplo, el sumador 314) se configura para añadir el bloque de transformada inversa 313 (es decir, un bloque residual reconstruido 313) al bloque de predicción 365, por ejemplo, para añadir un valor del muestra del bloque residual reconstruido 313 y un valor de muestra del bloque de predicción 365, para obtener un bloque reconstruido 315 en un dominio de muestras.

La unidad de filtro de bucle 320 (durante un bucle de codificado o después de un bucle de codificado) se configura para filtrar el bloque reconstruido 315 para obtener un bloque filtrado 321, para suavizar la transición de píxeles o mejorar la calidad de vídeo. En un ejemplo, la unidad de filtro de bucle 320 puede configurarse para realizar cualquier combinación de las tecnologías de filtrado descritas más adelante. La unidad de filtro de bucle 320 se destina a representar uno o más filtros de bucle, tal como un filtro de des-bloqueo, un filtro de desplazamiento adaptativo de muestra (SAO) u otros filtros, por ejemplo, un filtro bilateral o un filtro de bucle adaptativo (ALF) o un filtro de nitidez o filtros de suavizado o filtros colaborativos. Aunque la unidad de filtro de bucle 320 se muestra en la FIG. 3 como un filtro de bucle, en otra configuración, la unidad de filtro de bucle 320 puede implementarse como un filtro posterior al bucle.

Un bloque de vídeo decodificado 321 en un fotograma o imagen dados se almacena luego en el búfer de imágenes decodificadas 330, que almacena una imagen de referencia usada para compensación de movimiento posterior.

El decodificador 30 se configura para, por ejemplo, tener como salida la imagen decodificada 31, por ejemplo, a través de la salida 332, para presentación a un usuario o visualización por un usuario.

Se puede usar otra variante del decodificador de vídeo 30 para decodificar el flujo de bits comprimido. Por ejemplo, el decodificador 30 puede generar el flujo de bits de vídeo de salida sin la unidad de filtrado de bucle 320. Por ejemplo, un codificador no basado en transformada 30 puede cuantizar directamente de forma inversa una señal residual sin la unidad de procesamiento de transformada inversa 312 para algunos bloques o fotogramas. En otra implementación, el decodificador de vídeo 30 puede tener la unidad de cuantización inversa 310 y la unidad de procesamiento de transformada inversa 312 que se combinan en una sola unidad.

Específicamente, en esta realización de la presente invención, el decodificador 30 se configura para implementar un método de predicción de vector de movimiento descrito en las realizaciones siguientes.

Debe entenderse que otra variante estructural del decodificador de vídeo 30 puede configurarse para decodificar un flujo de bits de vídeo codificado. Por ejemplo, el decodificador de vídeo 30 puede generar un flujo de bits de vídeo de salida sin procesamiento por el filtro 320. Alternativamente, para algunos bloques de imagen o fotogramas de imagen, la unidad de decodificación entrópica 304 del decodificador de vídeo 30 no obtiene coeficientes cuantizados a través de la decodificación y, en consecuencia, no se requiere el

procesamiento por la unidad de cuantización inversa 310 y la unidad de procesamiento de transformada inversa 312. El filtro de bucle 320 es opcional, y en un caso de compresión sin pérdida, la unidad de cuantización inversa 310 y la unidad de procesamiento de transformada inversa 312 son opcionales. Debe entenderse que, según diferentes escenarios de aplicación, la unidad de interpredicción y la unidad de intrapredicción pueden habilitarse selectivamente.

Debe entenderse que en el codificador 20 y el decodificador 30 de esta solicitud, un resultado de procesamiento para un procedimiento puede ser enviado a un siguiente procedimiento después de ser procesado. Por ejemplo, después de un procedimiento tal como filtrado de interpolación, derivación de vector de movimiento, o filtrado de bucle, una operación tal como recorte o desplazamiento se realiza adicionalmente en un resultado de procesamiento del procedimiento correspondiente.

Por ejemplo, un vector de movimiento de un punto de control del bloque de imagen actual o un vector de movimiento de un subbloque del bloque de imagen actual derivado de un vector de movimiento de un bloque de codificado afín vecino pueden procesarse más a fondo. Esto no se limita en esta solicitud. Por ejemplo, un valor del vector de movimiento está restringido para estar dentro de un intervalo de profundidad de bits específico. Suponiendo que una profundidad de bits permitida del vector de movimiento es bitDepth, el valor del vector de movimiento varía de  $-2^{(bitDepth - 1)}$  a  $2^{(bitDepth - 1)} - 1$ , donde el símbolo " $\wedge$ " representa la potenciación. Si bitDepth es 16, el valor oscila entre -32768 y 32767. Si bitDepth es 18, el valor varía de -131072 a 131071. Por otro ejemplo, el valor del vector de movimiento (por ejemplo, los vectores de movimiento MV de cuatro subbloques 4x4 en un bloque de imagen 8 x 8) está restringido, de modo que una diferencia máxima entre las partes enteras de los MV de los cuatro subbloques 4 x 4 no excede N píxeles, por ejemplo, no excede un píxel.

Se pueden usar las siguientes dos maneras para restringir el valor del vector de movimiento para que esté dentro de un intervalo de profundidad de bits específico:

Manera 1: Se elimina el bit más significativo de desbordamiento del vector de movimiento:

$$ux = (vx + 2^{bitDepth}) \% 2^{bitDepth}$$

$$vx = (ux \geq 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux$$

$$uy = (vy + 2^{bitDepth}) \% 2^{bitDepth}$$

$$vy = (uy \geq 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy$$

vx representa un componente horizontal del vector de movimiento del bloque de imagen o del subbloque del bloque de imagen. vy representa un componente vertical del vector de movimiento del bloque de imagen o del subbloque del bloque de imagen. ux y uy son valores intermedios. bitDepth representa un bit de profundidad.

Por ejemplo, un valor de vx es -32769, y se obtiene 32767 según las fórmulas anteriores. Un valor se almacena en un ordenador en una representación de complemento a dos, una representación de complemento a dos de -32769 es 1,0111,1111,1111,1111 (17 bits), y procesamiento realizado por el ordenador para el desbordamiento es descartar el bit más significativo. Por lo tanto, un valor de vx es 0111,1111,1111,1111, es decir, 32767. Este valor es congruente con el resultado obtenido a través del procesamiento según las fórmulas.

Manera 2: El recorte se realiza en el vector de movimiento según las siguientes fórmulas:

$$vx = \text{Clip 3} (-2^{bitDepth-1}, 2^{bitDepth-1} - 1, vx)$$

$$vy = \text{Clip 3} (-2^{bitDepth-1}, 2^{bitDepth-1} - 1, vy)$$

vx representa un componente horizontal del vector de movimiento del bloque de imagen o el subbloque del bloque de imagen. vy representa un componente vertical del vector de movimiento del bloque de imagen o el subbloque del bloque de imagen. x, y, z respectivamente corresponden a tres valores de entrada de un proceso de sujeción MV Clip 3. Clip 3 se define para indicar recortar un valor de z en un intervalo [x, y].

$$\text{Clip 3}(x, y, z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{Si no} \end{cases}$$

La FIG. 4 es un diagrama esquemático de una estructura de un dispositivo de codificado de vídeo 400 (por ejemplo, un dispositivo de codificación de vídeo 400 o un dispositivo de decodificación de vídeo 400) según

una realización de la presente invención. El dispositivo de codificado de vídeo 400 es adecuado para implementar las realizaciones descritas como en esta memoria descriptiva. En una realización, el dispositivo de codificado de vídeo 400 puede ser un decodificador de vídeo (por ejemplo, el decodificador 30 en la FIG. 1A) o un codificador de vídeo (por ejemplo, el codificador 20 en la FIG. 1A). En otra realización, el dispositivo de codificado de vídeo 400 puede ser uno o más componentes del decodificador 30 en la FIG. 1A o el codificador 20 en la FIG. 1A.

El dispositivo de codificado de vídeo 400 comprende puertos de entrada 410 y una unidad de receptor (Rx) 420 para recibir datos; un procesador, una unidad lógica o unidad de procesamiento central (CPU) 430 para procesar datos; una unidad de transmisor (Tx) 440 y puertos de salida 450 para transmitir datos; y una memoria 460 para almacenar datos. El dispositivo de codificado de vídeo 400 puede incluir además un componente de conversión óptico a eléctrico y un componente eléctrico a óptico (EO) que se acoplan al puerto de entrada 410, la unidad de receptor 420, la unidad de transmisor 440 y el puerto de salida 450 para la salida o entrada de señales ópticas o eléctricas.

El procesador 430 se implementa usando hardware y software. El procesador 430 puede implementarse como uno o más chips de CPU, núcleos (por ejemplo, como un procesador multinúcleo), FPGA, ASIC y DSP. El procesador 430 se comunica con el puerto de entrada 410, la unidad de receptor 420, la unidad de transmisor 440, el puerto de salida 450 y la memoria 460. El procesador 430 incluye un módulo de codificado 470 (por ejemplo, un módulo de codificación 470 o un módulo de decodificación 470). El módulo de codificación/decodificación 470 implementa las realizaciones descritas en esta memoria descriptiva, para implementar el método de predicción de bloques de croma proporcionado en las realizaciones de la presente invención. Por ejemplo, el módulo de codificación/decodificación 470 implementa, procesa o proporciona diversas operaciones de codificado. Por lo tanto, el módulo de codificación/decodificación 470 proporciona una mejora sustancial a una función del dispositivo de codificado de vídeo 400 y efectúa una transformada del dispositivo de codificado de vídeo 400 a un estado diferente. De manera alternativa, el módulo de codificación/decodificación 470 se implementa como instrucciones almacenadas en la memoria 460 y ejecutadas por el procesador 430.

La memoria 460 incluye uno o más discos, unidades de cinta y unidades de estado sólido y puede usarse como dispositivo de almacenamiento de datos de desbordamiento (overflow), para almacenar programas cuando tales programas se seleccionan para su ejecución y para almacenar instrucciones y datos que se leen durante la ejecución del programa. La memoria 460 puede ser volátil o no volátil y puede ser una memoria de solo lectura (ROM), una memoria de acceso aleatorio (RAM), una memoria ternaria de contenido direccionable (TCAM), y/o una memoria estática de acceso aleatorio (SRAM).

La FIG. 5 es un diagrama de bloques simplificado de un aparato 500 que se puede usar como uno o ambos del dispositivo de fuente 12 y dispositivo de destino 14 de la FIG. 1A según una realización de ejemplo. El aparato 500 puede implementar las tecnologías de esta solicitud. En otras palabras, la FIG. 5 es un diagrama de bloques esquemático de una implementación de un dispositivo de codificación o un dispositivo de decodificación (un dispositivo de codificado 500 para abreviar) según una realización de esta solicitud. El dispositivo de codificado 500 puede incluir un procesador 510, una memoria 530 y un sistema de bus 550. El procesador y la memoria se conectan a través del sistema de bus. La memoria se configura para almacenar instrucciones. El procesador se configura para ejecutar las instrucciones almacenadas en la memoria. La memoria del dispositivo de codificado almacena el código de programa. El procesador puede invocar el código de programa almacenado en la memoria, para realizar diversos métodos de codificación o decodificación de vídeo descritos en esta solicitud, y en particular, diversos nuevos métodos de interpredicción. Para evitar la repetición, los detalles no se describen en esta memoria.

En esta realización de esta solicitud, el procesador 510 puede ser un conjunto central de procesamiento (CPU), o el procesador 510 puede ser otro procesador de propósito general, un procesador de señal digital (DSP), un circuito integrado específico de la aplicación (ASIC), una distribución de puertas programables en campo (FPGA), u otro dispositivo lógico programable, una puerta discreta o dispositivo lógico de transistores, componente de hardware discreto, o similar. El procesador de uso general puede ser un microprocesador, o el procesador puede ser cualquier procesador convencional o similar.

La memoria 530 puede incluir un dispositivo de memoria de solo lectura (ROM) o un dispositivo de memoria de acceso aleatorio (RAM). Cualquier otro tipo adecuado de dispositivo de almacenamiento se puede usar de manera alternativa como memoria 530. La memoria 530 puede incluir código y datos 531 a los que accede el procesador 510 a través del bus 550. La memoria 530 puede incluir además un sistema operativo 533 y un programa de aplicación 535. El programa de aplicación 535 incluye al menos un programa que permite al procesador 510 realizar el método de codificación o decodificación de vídeo descrito en esta solicitud (en particular, el método de predicción de vector de movimiento descrito en esta solicitud). Por ejemplo, los programas de aplicación 535 pueden incluir aplicaciones 1 a N, e incluye además una aplicación de codificación o decodificación de vídeo (denominada brevemente aplicación de codificado de vídeo) que se utiliza para realizar el método de codificación o decodificación de vídeo descrito en esta solicitud.

Además de un bus de datos, el sistema de bus 550 puede incluir además un bus de alimentación, un bus de control, un bus de señales de estado y similares. Sin embargo, para una mayor claridad de descripción, en la figura se marcan los diversos tipos de buses como el sistema de bus 550.

5 Opcionalmente, el dispositivo de codificado 500 puede incluir además uno o más dispositivos de salida, por ejemplo, una pantalla 570. En un ejemplo, la pantalla 570 puede ser una pantalla sensible al tacto que combina una pantalla con una unidad sensible al tacto que puede funcionar para detectar una entrada táctil. La pantalla 570 puede conectarse al procesador 510 a través del bus 550.

10 Tanto los aparatos o dispositivos que se muestran en la FIG. 4 como en la FIG. 5 pueden configurarse para realizar los métodos en las realizaciones de esta solicitud.

15 Como se ha descrito anteriormente, la interpredicción es una parte importante de un sistema de codificación y decodificación de vídeo.

En HEVC, se introducen los siguientes dos modos de interpredicción: un modo avanzado de predicción de vectores de movimiento (AMVP) y un modo de fusión.

20 En el modo AMVP, primero se construye una lista de vectores de movimiento de candidato basada en la información de movimiento de una unidad codificada que es espacialmente o temporalmente vecina a una unidad codificadora actual, y luego se determina un vector de movimiento óptimo a partir de la lista de vectores de movimiento de candidato y se utiliza como un predictor de vectores de movimiento (MVP) de la unidad codificadora actual. Un coste de distorsión de velocidad se calcula según una fórmula (1), donde  $J$  representa el coste de distorsión de velocidad (RD Cost),  $SAD$  representa una suma de diferencias absolutas (SAD) entre los valores de píxeles predichos y los valores de píxeles originales que se obtienen a través de la estimación de movimiento realizada mediante el uso de un predictor de vector de movimiento de candidato,  $R$  representa una tasa de bits y  $\lambda$  representa un multiplicador de Lagrange. Un lado de codificador transfiere, a un lado de decodificador, un valor de índice de un fotograma de referencia y un valor de índice que es del predictor de vector de movimiento seleccionado en función del coste de distorsión de velocidad y que está en la lista de vectores de movimiento de candidatos. Además, la búsqueda de movimiento se realiza en un dominio vecino centrado en MVP, para obtener un vector de movimiento real de la unidad codificadora actual. El lado de codificador transfiere una diferencia (diferencia de vector de movimiento, MVD) entre el MVP y el vector de movimiento real al lado de decodificador.

$$35 \quad J = SAD + \lambda R \quad (1)$$

40 En el modo de fusión, una lista de información de movimiento de candidato se construye primero en función de la información de movimiento de una unidad codificada que es espacialmente o temporalmente vecina a una unidad codificadora actual, la información de movimiento óptima se determina a partir de la lista de información de movimiento de candidato con base en un coste de distorsión de velocidad y se utiliza como información de movimiento de la unidad codificadora actual, y un valor de índice (denotado como un índice de fusión, el mismo a continuación) de una posición de la información de movimiento óptima en la lista de información de movimiento de candidato se transfiere al lado de decodificador. La información de movimiento de candidato espacial y temporal de la unidad codificadora actual se muestra en la FIG. 6. La información de movimiento de candidato espacial es de cinco bloques espacialmente vecinos ( $A_0$ ,  $A_1$ ,  $B_0$ ,  $B_1$  y  $B_2$ ). Si un bloque vecino no está disponible o está en modo de intracodificado o en modo de copia intrabloque, el bloque vecino no se añade a la lista de información de movimiento de candidato. La información de movimiento temporal candidato de la unidad codificadora actual se obtiene escalando un MV de un bloque colocado en un fotograma de referencia basado en recuentos de orden de imagen (POC) del fotograma de referencia y un fotograma actual. Determinar el bloque colocado incluye: primero determinar si un bloque en una posición  $T$  en el fotograma de referencia está disponible, y si el bloque en la posición  $T$  en el fotograma de referencia no está disponible, seleccionar un bloque en una posición  $C$  como el bloque colocado.

55 Durante la interpredicción en HEVC, la compensación de movimiento se realiza basándose en la suposición de que todos los píxeles de una unidad codificadora tienen la misma información de movimiento, para obtener valores de predicción de los píxeles de la unidad codificadora. Sin embargo, en la unidad codificadora, no todos los píxeles necesariamente tienen una misma función de movimiento. Por lo tanto, la predicción de todos los píxeles en la CU utilizando la misma información de movimiento puede reducir la precisión de la compensación de movimiento y aumentar aún más la información residual.

60 Para aumentar aún más la eficiencia de codificado, en algunas implementaciones factibles, la unidad codificadora se divide en al menos dos subunidades codificadoras, luego se deriva la información de movimiento de cada subunidad codificadora y se realiza la compensación de movimiento en función de la información de movimiento de las subunidades codificadoras. Esto mejora la precisión de predicción. Por ejemplo, se utiliza una tecnología de predicción de vectores de movimiento basada en unidades de

subcodificado (predicción de vectores de movimiento basada en Sub-CU, SMVP).

5 Según el SMVP, una unidad codificadora actual se divide en subunidades codificadoras que tienen un tamaño de  $M \times N$ , se deriva la información de movimiento de cada subunidad codificadora y luego se realiza la compensación de movimiento utilizando la información de movimiento de cada subunidad codificadora, para obtener un valor de predicción de la unidad codificadora actual.

10 Basado en la tecnología SMVP, se añade uno o dos tipos de información de movimiento de candidato, es decir, predicción avanzada de vector de movimiento temporal (ATMVP) y/o predicción de vector de movimiento espacio-temporal (STMVP), a la lista de información de movimiento de predicción candidato del modo de fusión. En consecuencia, la lista de información de movimiento de predicción de candidatos también se conoce como lista de candidatos de fusión basada en subbloques.

15 En algunas implementaciones factibles, la lista de candidatos de fusión basada en subbloques incluye uno o más de: un modo de predicción ATMVP, un modo de predicción basado en un modelo afinado (que incluye un método de predicción en el que se utiliza un vector de movimiento de punto de control heredado y/o un método de predicción en el que se utiliza un vector de movimiento de punto de control construido), un modo de interpredicción plano (PLANAR).

20 Como se muestra en la FIG. 7, según una tecnología ATMVP, primero se determina un fotograma de referencia colocado, luego se divide una unidad codificadora actual en subunidades codificadoras que tienen un tamaño de  $M \times N$ , se obtiene, escala y convierte en información de movimiento de la subunidad codificadora actual la información de movimiento de un píxel en una posición central de una subunidad codificadora que corresponde a cada subunidad codificadora actual y que está en el fotograma de referencia colocado. ATMVP también se conoce como predicción de vectores de movimiento temporal basada en subbloques (SbTMVP).

30 Según una tecnología STMVP, se obtiene información de movimiento en una posición espacialmente vecina superior, información de movimiento en una posición espacialmente vecina izquierda, e información de movimiento en una posición temporalmente vecina inferior derecha de cada subunidad de codificado actual, se calcula un promedio de las piezas de información de movimiento y se convierte en información de movimiento de la subunidad de codificado actual. Como se muestra en la FIG. 8, la unidad codificadora actual se divide en cuatro subunidades codificadoras: A, B, C y D. A se utiliza como ejemplo. La información de movimiento de la subunidad de codificado A se deriva utilizando información de movimiento en posiciones espacialmente vecinas c y b e información de movimiento en la posición D en el fotograma de referencia colocado.

40 Después de utilizar la tecnología SMVP, la información de movimiento de candidato ATMVP y la información de movimiento de candidato STMVP se añaden a la lista de información de movimiento de candidato del modo de fusión. En este caso, un proceso de codificación en el modo de fusión se convierte de la siguiente manera:

45 (1) Obtener secuencialmente información de movimiento en las posiciones espacialmente vecinas A1, B1, B0 y A0, comprobar la disponibilidad de información de movimiento en cada posición, opcionalmente, eliminar un elemento duplicado e insertar la información de movimiento no eliminada disponible en la lista de información de movimiento de candidato.

50 (2) Obtener información de movimiento correspondiente utilizando las tecnologías ATMVP y STMVP, comprobar la disponibilidad y eliminar un elemento duplicado, comprobar la disponibilidad de la información de movimiento obtenida, opcionalmente, eliminar un elemento duplicado e insertar la información de movimiento no eliminada disponible en la lista de información de movimiento de candidato.

55 (3) Cuando una longitud de la lista de información de movimiento de candidato sea menor que 6, obtener información de movimiento en la posición B2, comprobar la disponibilidad de la información de movimiento, opcionalmente, eliminar un elemento duplicado e insertar la información de movimiento en la lista de información de movimiento de candidato si la información de movimiento está disponible y no se elimina.

60 (4) Obtener información de movimiento del bloque correspondiente a la posición T en un fotograma codificado vecino (si la información de movimiento no existe, se obtiene información de movimiento de un bloque correspondiente a la posición C), realizar la escala en función de una relación POC entre el fotograma codificado vecino y el fotograma actual, e insertar información de movimiento escalado en la lista de información de movimiento de candidato.

65 (5) Si la longitud de la lista de información de movimiento de candidato es menor que 7, rellenar la información de movimiento, para obtener la lista de información de movimiento de candidato cuya longitud es 7, donde la información de movimiento rellena puede ser 0 u otra información de movimiento obtenida de una manera acordada previamente. Esto no está limitado.

5 (6) Atravesar cada elemento de información de movimiento de candidato en la lista de información de movimiento de candidato, realizar la compensación de movimiento y la reconstrucción para obtener un valor reconstruido, y luego determinar la información de movimiento de candidato con un coste de distorsión de velocidad más pequeño (RD cost) según un método de optimización de distorsión de velocidad (RDO), para obtener un índice de fusión correspondiente a la información de movimiento de candidato con un coste de RD más pequeño.

10 (7) Escribir el índice de fusión en un flujo de bits basado en la longitud de la lista de información de movimiento de candidato y transferir el flujo de bits al lado de decodificador.

15 Un proceso de realizar una predicción entre pares en una imagen a procesar actual mediante el uso de la tecnología ATMVP incluye principalmente: determinar un vector de movimiento de desplazamiento de un bloque a procesar actual en la imagen a procesar actual; determinar un subbloque colocado de un subbloque a procesar en una imagen de referencia colocada en función de una posición del subbloque a procesar en el bloque a procesar actual y el vector de movimiento de desplazamiento; determinar un vector de movimiento del subbloque a procesar actual con base en un vector de movimiento del subbloque a procesar; y realizar una predicción de compensación de movimiento en el subbloque a procesar en función del vector de movimiento del subbloque a procesar, para obtener un valor de píxel predicho del subbloque a procesar.

20 Específicamente, como se muestra en la FIG. 9, en una implementación factible, un método incluye las siguientes etapas.

S901: Obtener un vector de movimiento de desplazamiento.

25 El vector de movimiento de desplazamiento se utiliza para determinar una posición de un punto que corresponde a un punto de posición preestablecido en una CU actual (es decir, una CU actual a procesar, o también se puede denominar un bloque a procesar, un bloque a codificar, un bloque a decodificar o similar) y que está en una imagen colocada. Un vector de movimiento de un bloque espacialmente vecino de la CU actual se puede utilizar como vector de movimiento de desplazamiento. Una imagen colocada de un fotograma en el que se encuentra la CU actual puede obtenerse analizando un flujo de bits, o puede ser una imagen que tiene un valor preestablecido (el valor preestablecido es el mismo en un lado de codificador y un lado de decodificador). Por ejemplo, se supone que la imagen colocada es una imagen cuyo índice de fotograma de referencia es 0 en una lista de fotogramas de referencia de la CU actual. En algunas implementaciones factibles, también se puede determinar una posición de la CU actual en un bloque de imagen colocada en función del vector de movimiento de desplazamiento y el bloque de imagen colocada. Un bloque en la posición se puede denominar bloque colocado (bloque correspondiente/colocado).

40 Por ejemplo, el vector de movimiento de desplazamiento se puede obtener utilizando cualquiera de los siguientes métodos.

45 Método 1: Determinar si está disponible un vector de movimiento de A1 en la FIG. 6. Debe entenderse que, si el vector de movimiento está disponible, significa que el vector de movimiento existe y se puede obtener. Por ejemplo, el vector de movimiento del bloque vecino no está disponible cuando el bloque vecino no existe, el bloque vecino y el bloque actual no están en una misma región de codificado (por ejemplo, un segmento, un mosaico o un grupo de mosaicos), o se utiliza un modo de intrapredicción o un modo de copia intrabloque (copia intrabloque, IBC) para el bloque vecino. Por el contrario, el vector de movimiento del bloque vecino está disponible cuando se usa un modo de interpredicción para el bloque vecino, o cuando se usa un modo de interpredicción para el bloque vecino y el bloque actual están en una misma región de codificado.

50 Para el modo de copia intrabloque, se utiliza una herramienta de codificado intrabloque en un estándar extendido como el estándar de codificado de contenido de pantalla (screen content coding, SCC) del HEVC, y se utiliza principalmente para aumentar la eficiencia de codificado de un vídeo de contenido de pantalla. El modo IBC es un modo de codificado a nivel de bloque. En el lado de codificador, se utiliza un método de coincidencia de bloques para encontrar un vector de bloque o vector de movimiento óptimo para cada CU. El vector de movimiento en esta memoria se utiliza principalmente para representar un desplazamiento del bloque actual a un bloque de referencia, y también se conoce como vector de desplazamiento. El bloque de referencia es un bloque reconstruido en una imagen actual. El modo IBC puede considerarse como un tercer modo de predicción que no sea el modo de intrapredicción o el modo de interpredicción. Para ahorrar espacio de almacenamiento y reducir la complejidad de un decodificador, en algunas implementaciones, el modo IBC permite que solo se utilice para la predicción una parte reconstruida en una región predefinida de una CTU actual.

65 Cuando A1 no está disponible, se puede utilizar un vector de movimiento cero como vector de movimiento de desplazamiento de la CU actual.

Se supone que A1 puede tener un primer vector de movimiento de dirección que se basa en una primera lista de fotogramas de referencia 0 y un segundo vector de movimiento de dirección que se basa en una segunda lista de fotogramas de referencia 1.

5 Cuando A1 está disponible, si A1 cumple todas las condiciones siguientes (1) a (5), el segundo vector de movimiento de dirección que es de A1 y que se basa en la lista 1 se utiliza como vector de movimiento de desplazamiento de la CU actual. Las condiciones son las siguientes.

10 (1) Se utiliza un fotograma de referencia en la lista 1 para A1 para la predicción.

(2) El fotograma de referencia en la lista 1 de A1 utilizado para la predicción es el mismo que una imagen colocada de un fotograma de imagen en el que se encuentra la CU actual.

15 Por ejemplo, se puede determinar si un POC del fotograma de referencia en la lista 1 es el mismo que un POC de la imagen colocada del fotograma de imagen en el que se encuentra la CU actual, donde la información que representa la imagen colocada se puede obtener analizando el flujo de bits.

20 (3) Se utiliza una estructura de codificado de baja latencia, para ser específicos, los fotogramas de referencia de una imagen a codificar/decodificar/procesar se muestran antes de la imagen a codificar/decodificar/procesar.

(4) Un tipo de imagen de una imagen en la que se encuentra la CU actual es una imagen B, un mosaico en el que se encuentra la CU actual es un mosaico B, o un grupo de mosaicos en los que se encuentra la CU actual es un grupo de mosaicos B.

25 (5) Una imagen colocada de la imagen en la que se encuentra la CU actual se obtiene de la lista 1. Por ejemplo, un valor de un elemento de sintaxis `collocated_from_10_flag` es 0.

30 De lo contrario, cuando A1 está disponible pero no cumple al menos una de las condiciones anteriores, si A1 cumple las siguientes condiciones (6) y (7), el primer vector de movimiento de dirección que es de A1 y que se basa en la lista 0 se utiliza como vector de movimiento de desplazamiento de la CU actual. Las condiciones son las siguientes.

35 (6) Se utiliza un fotograma de referencia en la lista 0 para A1 para la predicción.

(7) El fotograma de referencia en la lista 0 de A1 utilizado para la predicción es el mismo que la imagen colocada del fotograma de imagen en el que se encuentra la CU actual.

40 Método 2: Encontrar un vector de movimiento de un primer bloque vecino disponible en una secuencia de A1, B 1, B0 y A0 en la FIG. 6. Si un fotograma de referencia del bloque vecino es una imagen colocada de un fotograma en el que se encuentra la CU actual, es decir, el vector de movimiento encontrado apunta a la imagen colocada, el vector de movimiento del bloque vecino se utiliza como vector de movimiento de desplazamiento de la CU actual. De lo contrario, si un fotograma de referencia del bloque vecino no es una imagen colocada de un fotograma en el que se encuentra la CU actual, en una implementación factible, se puede usar un vector de movimiento cero como el vector de movimiento de desplazamiento de la CU actual. En otra implementación factible, el vector de movimiento del primer bloque vecino disponible se escala en función de un POC de la imagen colocada, un POC de una imagen en la que se encuentra la CU actual y un POC del fotograma de referencia del primer bloque vecino disponible, de modo que un vector de movimiento escalado apunta a la imagen colocada; y el vector de movimiento escalado se utiliza como vector de movimiento de desplazamiento de la CU actual. Para un método de escalado específico, se refiere a un método para obtener un vector de movimiento temporal en la tecnología convencional, o un método de escalado en la etapa 1005 en esta realización. Los detalles no se describen.

55 Debe entenderse que, cuando el vector de movimiento de desplazamiento es el vector de movimiento cero, un bloque de imagen que está en la imagen colocada y que está en la misma posición que la CU actual es un bloque colocado de la CU actual en la imagen colocada.

60 Cuando el vector de movimiento de desplazamiento de la CU actual que cumple con la condición no se puede obtener mediante el método anterior, un vector de movimiento de un subbloque de la CU actual no se obtiene mediante el uso de un modo de predicción ATMVP.

S902: Determinar, en función del vector de movimiento de desplazamiento, si el modo ATMVP está disponible.

65 Específicamente, se supone que un bloque de imagen en el que el punto que corresponde al punto de posición preestablecido en la CU actual y que está en la imagen colocada es un subbloque S, y una posición de coordenadas del subbloque S es  $(x_{col}, y_{col})$ . Por ejemplo,

$$\begin{cases} x_{col} = x + \frac{W}{2} + x_{off} \\ y_{col} = y + \frac{H}{2} + y_{off} \end{cases} \quad (2)$$

5 (x, y) representa las coordenadas de una esquina superior izquierda de la CU actual, W representa la anchura de la CU actual, H representa la altura de la CU actual y (x<sub>off</sub>, y<sub>off</sub>) representa el vector de movimiento de desplazamiento.

10 Cuando un modo de predicción del subbloque S es el modo de intrapredicción o el modo de copia intrabloque, se determina que el modo ATMVP no está disponible, y no se realizan la etapa 902 y las etapas posteriores.

15 Cuando un modo de predicción del subbloque S es el modo de interpredicción, se determina que el modo ATMVP está disponible. Además, se obtiene información de movimiento del subbloque S, para ser específicos, información de movimiento correspondiente a la posición de coordenadas (x<sub>col</sub>, y<sub>col</sub>), y se determina como un vector de movimiento predeterminado inicial.

20 El vector de movimiento predeterminado inicial MV se escala, para obtener un vector de movimiento predeterminado (MV) de un subbloque a procesar.

25 Por ejemplo, como se muestra en la FIG. 10, el MV se puede obtener según una fórmula (3):

$$MV_s = \frac{CurPoc - CurRefPoc}{ColPoc - ColRefPoc} \times MV \quad (3)$$

30 CurPoc representa un POC del fotograma en el que se encuentra la CU actual, ColPoc representa el POC de la imagen colocada, CurRefPoc representa un POC del fotograma de referencia de la CU actual y ColRefPoc representa un POC de un fotograma de referencia del subbloque S.

35 Debe entenderse que el MV incluye un vector de movimiento MVx en una dirección horizontal y un vector de movimiento MVy en una dirección vertical, y un vector de movimiento escalado MVsx en la dirección horizontal y un vector de movimiento escalado MVsy en la dirección vertical pueden obtenerse por separado a través del cálculo según la fórmula anterior.

40 S903: Determinar la información de movimiento del subbloque a procesar en función de la información de movimiento de un subbloque colocado.

45 Por ejemplo, como se muestra en la FIG. 11, el bloque a procesar, es decir, la CU actual, se encuentra en la imagen actual, el bloque a procesar incluye cuatro subbloques y un subbloque a procesar es un subbloque superior izquierdo del bloque a procesar. En este caso, un subbloque en una posición superior izquierda en la imagen colocada puede determinarse como un subbloque colocado del subbloque a procesar. Debe entenderse que pueden existir diferentes subbloques colocados en una forma de un bloque colocados en su conjunto, o pueden existir en una forma de subbloques separados.

50 Por ejemplo, se puede obtener información de movimiento correspondiente a una posición central geométrica del subbloque colocado. Específicamente, una posición central (x<sub>(i,j)</sub>, y<sub>(i,j)</sub>) de un subbloque colocado de un subbloque (i, j)-ésimo a procesar (es decir, un subbloque que es un i-ésimo subbloque de izquierda a derecha y un j-ésimo subbloque de arriba a abajo) de la CU actual puede obtenerse según una fórmula (4).

$$\begin{cases} x_{(i,j)} = x + M \times i + \frac{M}{2} + x_{off} \\ y_{(i,j)} = y + N \times j + \frac{N}{2} + y_{off} \end{cases} \quad (4)$$

55 (x, y) representa las coordenadas de una esquina superior izquierda de la CU actual, M representa la anchura del subbloque a procesar y H representa la altura del subbloque a procesar.

Se determina un modo de predicción de un bloque de imagen en el que se encuentra la posición central.

60 Cuando el modo de predicción del bloque de imagen es el modo de interpredicción, se dispone de un vector de movimiento en la posición central y se obtiene el vector de movimiento en la posición. El procesamiento de escala se realiza en el vector de movimiento basado en una relación temporal entre un fotograma de imagen en el que se encuentra el subbloque a procesar y un fotograma de imagen en el que se encuentra el subbloque colocado, para obtener un vector de movimiento del subbloque a procesar. Un método de procesamiento de escala es similar al de la fórmula (2). Por ejemplo,

$$MV_R = \frac{CurPoc - CurRefPoc}{ColPoc - ColRefPoc} \times MV \quad (5)$$

5 MV representa el vector de movimiento en la posición central, y  $MV_R$  representa el vector de movimiento del subbloque a procesar.

10 En una implementación factible, el POC  $CurRefPoc$  del fotograma de referencia de la CU actual puede preconfigurarse en un POC de un fotograma de referencia cuyo índice de fotograma de referencia es 0 en la lista de imágenes de referencia del fotograma en el que se encuentra la CU actual.

15 Debe entenderse que  $CurRefPoc$  puede ser, alternativamente, otro fotograma de referencia en la lista de imágenes de referencia del fotograma en el que se encuentra la CU actual, y esto no está limitado.

20 Cuando el modo de predicción del bloque de imagen es el modo de intrapredicción o el modo de copia intrabloque, el vector de movimiento en la posición central no está disponible, y el vector de movimiento predeterminado (MV) del subbloque a procesar que se determina en la etapa 5902 se utiliza como vector de movimiento del subbloque a procesar.

25 S904: Realizar una compensación de movimiento basada en la información de movimiento del subbloque de la CU actual, para obtener un valor de píxel previsto de la CU actual.

30 Para cada subbloque, la compensación de movimiento se realiza en función del vector de movimiento determinado en la etapa 903 y el fotograma de referencia de la imagen en la que se encuentra la CU actual, por ejemplo, el vector de movimiento  $MV_R$  y el fotograma de referencia  $CurRefPoc$ , para obtener un valor de píxel previsto del subbloque. Para un proceso de compensación de movimiento, consulte la descripción anterior o cualquier solución de mejora en la tecnología convencional. No se describen nuevamente detalles.

35 La CU actual incluye subbloques. Por lo tanto, después de obtener un valor de píxel predicho de cada subbloque de la manera anterior, se obtiene el valor de píxel predicho de la CU actual.

40 En esta implementación, se obtiene el vector de movimiento de cada subbloque, de modo que se puede reflejar una situación de movimiento más compleja dentro del bloque a procesar. Esto mejora la precisión del vector de movimiento y también aumenta la eficiencia de codificado. Sin embargo, cuando la información de movimiento del subbloque colocado no está disponible, es necesario calcular la información de movimiento predeterminada. Esto afecta a la velocidad de codificado.

45 Como se muestra en la FIG. 12, en otra implementación factible, un bloque a procesar incluye uno o más subbloques, y la información de identificación de una imagen colocada del bloque a procesar se obtiene analizando un flujo de bits. Cuando un vector de movimiento de un bloque espacialmente vecino en una posición preestablecida del bloque a procesar está disponible y un fotograma de referencia correspondiente al vector de movimiento es una imagen colocada, el vector de movimiento se determina como un vector de desplazamiento temporal. Una posición de un subbloque colocado del subbloque del bloque a procesar se determina en la imagen colocada en función del vector de desplazamiento temporal. Se determina si está disponible un vector de movimiento del subbloque colocado, y un vector de movimiento del subbloque del bloque a procesar se determina en función del vector de movimiento del subbloque colocado. El método de interpredicción incluye específicamente las siguientes etapas:

50 S1201: Determinar el vector de desplazamiento temporal del bloque a procesar en función del bloque espacialmente vecino del bloque a procesar.

55 El vector de desplazamiento temporal se utiliza para determinar el subbloque colocado del subbloque del bloque a procesar.

60 Debe entenderse que, por ejemplo, como se muestra en la FIG. 11, el bloque a procesar incluye cuatro subbloques a procesar, y un subbloque colocado de cada subbloque del bloque a procesar se determina en la imagen colocada (que se denota como una imagen objetivo, es decir, una imagen en la que se encuentra el subbloque colocado) en función de una posición del subbloque del bloque a procesar en una imagen actual y el vector de desplazamiento temporal (que se denota como vector de movimiento de desplazamiento en la FIG.).

Se obtiene un índice que es de un fotograma de imagen (la imagen colocada) en el que se encuentra el subbloque colocado y que está en una lista de fotogramas de referencia del bloque espacialmente vecino del bloque a procesar analizando el flujo de bits. Para ser específico, un lado de decodificador puede determinar la imagen colocada analizando la información correspondiente en el flujo de bits; y un lado de codificador puede

determinar, de una manera de selección RDO, un fotograma de imagen con un rendimiento óptimo como la imagen colocada, o especificar un fotograma como la imagen colocada, y escribir información de indicación de la imagen colocada en el flujo de bits. Alternativamente, la imagen colocada puede preconfigurarse por el lado de codificador y el lado de decodificador según un protocolo.

5

En una implementación factible, la etapa 1201 puede ser específicamente la siguiente:

comprobar secuencialmente, en un orden preestablecido, si los vectores de movimiento de bloques espacialmente vecinos en una pluralidad de primeras posiciones preestablecidas están disponibles, hasta que se obtenga un vector de movimiento de un primer bloque espacialmente vecino cuyo vector de movimiento está disponible en el orden preestablecido; y

10

utilizar, como vector de desplazamiento temporal, el vector de movimiento del primer bloque espacialmente vecino cuyo vector de movimiento está disponible en el orden preestablecido.

15

Un segundo vector de movimiento preestablecido se utiliza como el vector de desplazamiento temporal cuando los vectores de movimiento de los bloques espacialmente vecinos en la pluralidad de primeras posiciones preestablecidas no están disponibles.

Por ejemplo, como se muestra en la FIG. 6, se puede comprobar secuencialmente si están disponibles los vectores de movimiento de los bloques espacialmente vecinos A1, B 1, B0 y A0 del bloque a procesar, hasta que se encuentre un primer bloque espacialmente vecino cuyo vector de movimiento esté disponible. Se supone que el primer bloque espacialmente vecino es B0. En este caso, la comprobación se detiene y se utiliza un vector de movimiento de B0 como vector de desplazamiento temporal.

20

En una implementación factible, el procesamiento de escala se puede realizar adicionalmente en el vector de movimiento de B0 con base en una relación temporal entre un fotograma de referencia de B0, un fotograma de imagen en el que se encuentra el bloque a procesar y la imagen colocada del bloque a procesar, de modo que un vector de movimiento escalado utiliza la imagen colocada como fotograma de referencia.

25

El segundo vector de movimiento preestablecido, es decir, un vector de movimiento cero, se puede utilizar como vector de desplazamiento temporal cuando los vectores de movimiento de los bloques espacialmente vecinos no están disponibles.

Debe entenderse que el bloque espacialmente vecino en la primera posición preestablecida se preestablece por el lado de codificador y el lado de decodificador según un protocolo o se determina por el lado de codificador y el lado de decodificador en función de un elemento de sintaxis de nivel superior. Esto no está limitado en esta realización de esta solicitud.

Una condición de que el vector de movimiento del bloque espacialmente vecino no está disponible incluye uno o una combinación de los siguientes elementos: El bloque espacialmente vecino no se codifica/decodifica (si el método de predicción se implementa en el lado de codificador, el bloque espacialmente vecino no se codifica, o si el método de predicción se implementa en el lado de decodificador, el bloque espacialmente vecino no se decodifica); se utiliza un modo de interpretación o un modo de copia intrabloque para el bloque espacialmente vecino; el bloque espacialmente vecino no existe; o el bloque espacialmente vecino y el bloque a procesar se encuentran en diferentes regiones de codificado.

45

Por ejemplo, la región de codificado incluye una imagen, una segmento, un mosaico o un grupo de mosaicos.

En otra implementación factible, la etapa 1201 puede ser, alternativamente, específicamente la siguiente:

obtener un vector de movimiento y un fotograma de referencia de un bloque espacialmente vecino en una segunda posición preestablecida, donde el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida está disponible; y

55

utilizar el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida como vector de desplazamiento temporal.

Se utiliza un tercer vector de movimiento preestablecido como vector de desplazamiento temporal cuando el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida no está disponible.

60

En algunas implementaciones factibles, el bloque espacialmente vecino en la segunda posición preestablecida cumple además una condición de que el fotograma de referencia del bloque espacialmente vecino es el mismo que en el que se encuentra la imagen colocada.

65

Por ejemplo, como se muestra en la FIG. 6, se supone que el bloque espacialmente vecino en la segunda posición preestablecida es A1.

- 5 Debe entenderse que, el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida incluye un primer vector de movimiento de dirección que se basa en una primera lista de fotogramas de referencia, y el fotograma de referencia del bloque espacialmente vecino en la segunda posición preestablecida incluye un primer fotograma de referencia de dirección correspondiente al primer vector de movimiento de dirección.
- 10 En una implementación factible, el uso del vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida como el vector de desplazamiento temporal es específicamente: usar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado; o usar el tercer vector de movimiento preestablecido como el vector de desplazamiento temporal
- 15 cuando el primer fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado.

Por ejemplo, el tercer vector de movimiento preestablecido es un vector de movimiento cero.

- 20 Debe entenderse que, cuando se utiliza la predicción bidireccional para el bloque espacialmente vecino en la segunda posición preestablecida, el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida incluye además un segundo vector de movimiento de dirección que se basa en una segunda lista de fotogramas de referencia, y el fotograma de referencia del bloque espacialmente vecino en la segunda posición preestablecida incluye un segundo fotograma de referencia de dirección correspondiente al
- 25 segundo vector de movimiento de dirección.

- En otra implementación factible, el uso del vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida como el vector de desplazamiento temporal es específicamente: usar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque
- 30 colocado; o cuando el primer fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado, determinar si el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado; y usar el segundo vector de movimiento de dirección como vector de desplazamiento temporal cuando el segundo fotograma de referencia de imagen es diferente del fotograma de imagen colocada; o usar el tercer vector de movimiento preestablecido como el vector de desplazamiento temporal cuando el segundo fotograma de referencia de imagen es diferente del fotograma de imagen en el que se encuentra el subbloque colocado. Opcionalmente, antes de determinar si el segundo fotograma de referencia de dirección es el mismo que el fotograma de
- 35 imagen en el que se encuentra el subbloque colocado, el método incluye además: determinar si un tipo de una región de codificado en la que se ubica el bloque espacialmente vecino es un tipo B, es decir, si la región de codificado es un fotograma B, un mosaico B, un segmento B o un grupo de mosaicos B.
- 40

- En otra implementación factible, cuando se utiliza la predicción bidireccional para el bloque espacialmente vecino en la segunda posición preestablecida, el uso del vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida como el vector de desplazamiento temporal es específicamente: cuando el fotograma de imagen en el que se encuentra el subbloque colocado se obtiene de la segunda lista de fotogramas de referencia, determinar si el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado, y usar el segundo vector de movimiento de dirección como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el primer subbloque colocado, o usar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado y el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que encuentra el subbloque colocado; o cuando el fotograma de imagen en el que se encuentra el subbloque colocado se obtiene de la primera lista de fotogramas de referencia, determinar si el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado, y utilizar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado, o utilizar el segundo vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado y el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado. Además, el tercer vector de movimiento preestablecido se utiliza como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado y el primer fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque
- 45
- 50
- 55
- 60
- 65

En otra implementación factible, cuando se utiliza la predicción bidireccional para el bloque espacialmente vecino en la segunda posición preestablecida, el uso del vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida como el vector de desplazamiento temporal es específicamente: cuando el fotograma de imagen en el que se encuentra el subbloque colocado se obtiene de la segunda lista de fotogramas de referencia, y todos los fotogramas de referencia en una lista de fotogramas de referencia del bloque a procesar se muestran antes del fotograma de imagen en el que se encuentra el bloque a procesar, determinar si el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado, y usar el segundo vector de movimiento de dirección como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el primer subbloque colocado, o usar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado y el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que encuentra el subbloque colocado; o cuando el fotograma de imagen en el que se encuentra el subbloque colocado se obtiene de la primera lista de fotogramas de referencia, o al menos un fotograma de referencia en una lista de fotogramas de referencia del bloque a procesar se muestra después del fotograma de imagen en el que se encuentra el bloque a procesar, determinar si el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado, y utilizar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado, o utilizar el segundo vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado y el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado. Además, el tercer vector de movimiento preestablecido se utiliza como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado y el primer fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado.

Por ejemplo, se puede determinar si el fotograma de imagen en el que se encuentra el subbloque colocado se obtiene de la primera/segunda lista de fotogramas de referencia analizando un elemento de sintaxis `collocated_from_10_flag` en el flujo de bits. Específicamente, `collocated_from_10_flag` igual a 1 especifica que el fotograma de imagen (la imagen colocada) en el que se encuentra el subbloque colocado se obtiene de la primera lista de fotogramas de referencia; y `collocated_from_10_flag` igual a 0 especifica que el fotograma de imagen en el que se encuentra el subbloque colocado se obtiene de la segunda lista de fotogramas de referencia. Cuando el flujo de bits no lleva información sobre `collocated_from_10_flag`, el fotograma de imagen en el que se encuentra el subbloque colocado se obtiene de la primera lista de fotogramas de referencia de forma predeterminada.

Por ejemplo, todos los fotogramas de referencia de la lista de fotogramas de referencia del bloque a procesar se muestran antes del fotograma de imagen en el que se encuentra el bloque a procesar, es decir, se utiliza una estructura de codificado de fotogramas de baja latencia. En esta estructura de codificado, cuando se codifica cada fotograma de la imagen, todos los fotogramas de referencia utilizados se muestran antes de un fotograma actual que se va a codificar. En consecuencia, en el lado de decodificador, todos los fotogramas de referencia utilizados se muestran antes de un fotograma actual a decodificar.

Debe entenderse que, cuando el tipo de la región de codificado en la que se encuentra el bloque espacialmente vecino no es el tipo B, la predicción bidireccional no se utiliza para el bloque espacialmente vecino, y una implementación en la que se utiliza la predicción bidireccional puede no lograr un buen efecto técnico. Por lo tanto, opcionalmente, antes de la implementación anterior en la que se realiza la predicción bidireccional, se puede determinar si la región de codificado en la que se encuentra el bloque espacialmente vecino es del tipo B.

S1202: Determinar si está disponible un vector de movimiento correspondiente a una posición intrabloque preestablecida del subbloque colocado.

Una posición de un subbloque colocado de un subbloque (que se conoce como un subbloque de ejemplo) en el bloque a procesar se puede determinar en la imagen colocada en función de las coordenadas de posición del subbloque del bloque a procesar y el vector de desplazamiento temporal que se determina en la etapa S1201. Por ejemplo, la posición del subbloque colocado se puede obtener según la fórmula (4) en esta memoria descriptiva. Específicamente, con referencia a la fórmula (4),  $x$  e  $y$ , respectivamente, representan una coordenada horizontal y una coordenada vertical de una esquina superior izquierda del bloque a procesar;  $i$  y  $j$  representan que el subbloque de ejemplo es un subbloque que está dispuesto en el bloque a procesar y que es un  $i$ -ésimo subbloque de izquierda a derecha y un  $j$ -ésimo subbloque de arriba a abajo;  $x_{off}$  e  $y_{off}$  respectivamente representan una coordenada horizontal y una coordenada vertical del vector de compensación temporal;  $M$  y  $N$  respectivamente representan la anchura y la altura del subbloque; y  $x_{(i,j)}$  y  $y_{(i,j)}$  representan las coordenadas

de posición del subbloque colocado del ejemplo (que se conoce como subbloque colocado para abreviar).

5 Debe entenderse que  $M/2$  y  $N/2$  en la fórmula (4) representan que la posición intrabloque preestablecida es una posición central geométrica del subbloque colocado. La posición intrabloque preestablecida puede ser alternativamente otra posición intrabloque tal como una esquina superior izquierda del subbloque colocado. Esto no está limitado.

10 El vector de movimiento correspondiente a la posición intrabloque preestablecida del subbloque colocado también se puede utilizar como vector de movimiento del subbloque colocado.

15 Una unidad de predicción en la imagen colocada en la que las coordenadas de posición ( $X_{(i,j)}$ ,  $Y_{(i,j)}$ ) se encuentran localizados puede ser determinado con base en las coordenadas de posición, y si el vector de movimiento correspondiente a la posición intrabloque preestablecida del subbloque colocado está disponible puede ser determinado en función de la información de predicción de la unidad de predicción.

20 Debe entenderse que la unidad de predicción es un resultado obtenido después de que la imagen colocada se codifica realmente, y puede ser incongruente con el subbloque colocado.

25 Por ejemplo, cuando un modo de predicción de la unidad de predicción es un modo de interpredicción, el vector de movimiento correspondiente a la posición intrabloque preestablecida está disponible; o cuando un modo de predicción de la unidad de predicción es un modo de intrapredicción o un modo de copia intrabloque, el vector de movimiento correspondiente a la posición intrabloque preestablecida no está disponible.

30 En una implementación factible, se puede comprobar la información del modo de predicción de la unidad de predicción y se determina, en función de la información de modo de predicción, que el modo de predicción de la unidad de predicción es el modo de intrapredicción, el modo de interpredicción, el modo de copia interna del bloque u otro modo.

35 En otra implementación factible, se puede comprobar la información de movimiento de la unidad de predicción, por ejemplo, se puede comprobar una dirección de predicción. Cuando un indicador de dirección de predicción  $\text{predFlagL0}$  y/o el  $\text{predFlagL1}$  es 1, el modo de predicción es el modo de interpredicción. De lo contrario, el modo de predicción es el modo de intrapredicción u otro modo de predicción en el que el vector de movimiento no está disponible.

40 S1203: Determinar el vector de movimiento del subbloque del bloque a procesar en función del vector de movimiento del subbloque colocado.

45 El vector de movimiento del subbloque del bloque a procesar se obtiene en función de un primer vector de movimiento preestablecido cuando el vector de movimiento del subbloque colocado no está disponible.

50 Cuando el modo de predicción de la unidad de predicción es el modo de interpredicción, el vector de movimiento correspondiente a la posición intrabloque preestablecida está disponible, y el vector de movimiento del subbloque del bloque a procesar se obtiene en función del vector de movimiento correspondiente a la posición intrabloque preestablecida.

55 Específicamente, se puede realizar el procesamiento de escala, en función de una relación de una primera diferencia de distancia temporal a una segunda diferencia de distancia temporal, en el vector de movimiento correspondiente a la posición de intrabloque preestablecida, para obtener el vector de movimiento del subbloque del bloque a procesar, donde la primera diferencia de distancia temporal es una diferencia de recuento de orden de imagen entre el fotograma de imagen en el que se encuentra el bloque a procesar y un fotograma de referencia del bloque a procesar, y la segunda diferencia de distancia temporal es una diferencia de recuento de orden de imagen entre el fotograma de imagen en el que se encuentra el subbloque colocado y un fotograma de referencia del subbloque colocado. Para un procedimiento de cálculo específico para el procesamiento de escala, por ejemplo, consulte la fórmula (3) en esta memoria descriptiva. No se describen nuevamente detalles.

60 Se obtiene un índice del fotograma de referencia del bloque a procesar en la lista de fotogramas de referencia del bloque a procesar analizando el flujo de bits. Para ser específico, el lado de decodificador puede determinar el fotograma de referencia del bloque a procesar analizando la información correspondiente en el flujo de bits; y el lado de codificador puede determinar, de una manera de selección RDO, un fotograma de imagen con un rendimiento óptimo como el fotograma de referencia del bloque a procesar, o especificar un fotograma como el fotograma de referencia del bloque a procesar, y escribir información de indicación del fotograma de referencia del bloque a procesar en el flujo de bits. Alternativamente, el fotograma de referencia del bloque a procesar puede preconfigurarse por el lado de codificador y el lado de decodificador según un protocolo. Por ejemplo, el índice del fotograma de referencia del bloque a procesar en la lista de fotogramas de referencia del bloque a procesar es 0.

5 Debe entenderse que el vector de movimiento correspondiente a la posición intrabloque preestablecida puede obtenerse directamente de una unidad de almacenamiento de vector de movimiento correspondiente a la posición, puede obtenerse de una unidad de almacenamiento de vector de movimiento correspondiente a una posición vecina, o puede obtenerse a través del filtrado de interpolación basado en un vector de movimiento en una unidad de almacenamiento de vector de movimiento correspondiente a una posición vecina. Esto no está limitado.

10 Cuando el modo de predicción de la unidad de predicción es el modo de intrapredicción o el modo de copia intrabloque, el vector de movimiento correspondiente a la posición intrabloque preestablecida no está disponible, y el vector de movimiento del subbloque del bloque a procesar se obtiene en función del primer vector de movimiento preestablecido.

15 En una implementación factible, el vector de movimiento del subbloque incluye un primer vector de movimiento de subbloque de dirección que se basa en la primera lista de fotogramas de referencia y/o un segundo vector de movimiento de subbloque de dirección que se basa en la segunda lista de fotogramas de referencia; y cuando el vector de movimiento correspondiente a la posición de intrabloque preestablecida no está disponible, que el vector de movimiento del subbloque del bloque a procesar se obtiene en función del primer vector de movimiento preestablecido es específicamente: determinar que la predicción unidireccional basada en el primer vector de movimiento de subbloque de dirección se utiliza para el subbloque del bloque a procesar, y obtener el primer vector de movimiento de subbloque de dirección del subbloque del subbloque a procesar en función del primer vector de movimiento preestablecido; o determinar esa segunda predicción unidireccional en función del segundo vector de movimiento de subbloque de dirección se utiliza para el subbloque del bloque a procesar y para obtener el segundo vector de movimiento de subbloque de dirección del subbloque del bloque a procesar en función del primer vector de movimiento preestablecido.

30 En otra implementación factible, el vector de movimiento del subbloque incluye un primer vector de movimiento de subbloque de dirección que se basa en la primera lista de fotogramas de referencia y un segundo vector de movimiento de subbloque de dirección que se basa en la segunda lista de fotogramas de referencia; y cuando el vector de movimiento correspondiente a la posición de intrabloque preestablecida no está disponible, que el vector de movimiento del subbloque del bloque a procesar se obtiene en función del primer vector de movimiento preestablecido es específicamente: cuando un tipo de predicción de una región de codificado en la que se encuentra el bloque a procesar es predicción tipo B, determinar que la predicción bidireccional se utiliza para el subbloque del bloque a procesar, y obtener por separado el primer vector de movimiento de subbloque de dirección del subbloque del subbloque a procesar y el segundo vector de movimiento de subbloque de dirección del subbloque del subbloque del bloque a procesar en función del primer vector de movimiento preestablecido; o cuando un tipo de predicción de una región de codificado en la que se encuentra el bloque a procesar es predicción tipo P, determinar que la predicción unidireccional se utiliza para el subbloque del bloque a procesar y obtener el primer vector de movimiento de subbloque de dirección del subbloque del bloque a procesar en función del primer vector de movimiento preestablecido.

Por ejemplo, la región de codificado incluye una imagen, un segmento, un mosaico o un grupo de mosaicos.

45 En una implementación factible, que el vector de movimiento del subbloque del bloque a procesar se obtenga en función de un primer vector de movimiento preestablecido incluye: usar el primer vector de movimiento preestablecido como vector de movimiento del subbloque del bloque a procesar.

50 En una implementación factible, el primer vector de movimiento preestablecido es un vector de movimiento cero.

Por ejemplo, un indicador de predicción de primera dirección  $predFlagL0$  puede establecerse en 1, un indicador de predicción de segunda dirección  $predFlagL1$  puede establecerse en 0 y el primer vector de movimiento de subbloque de dirección  $mvL0$  puede establecerse en (0, 0).

55 Como alternativa, un indicador de predicción de primera dirección  $predFlagL0$  puede establecerse en 0, un indicador de predicción de segunda dirección  $predFlagL1$  puede establecerse en 1 y el segundo vector de movimiento de subbloque de dirección  $mvL1$  puede establecerse en (0, 0).

60 Alternativamente, cuando una región de codificado en la que se ubica el subbloque de ejemplo es un fotograma B, un segmento B, un mosaico B o un grupo de mosaicos B, se puede establecer un primer indicador de predicción de dirección  $predFlagL0$  en 1, un segundo indicador de predicción de dirección  $predFlagL1$  en 1, el primer vector de movimiento de subbloque de dirección  $mvL0$  en (0, 0) y el segundo vector de movimiento de subbloque de dirección  $mvL1$  puede establecerse en (0, 0). De lo contrario (cuando una región de codificado en la que se encuentra el subbloque de ejemplo no corresponde a la región de predicción bidireccional anterior), el primer indicador de predicción de dirección  $predFlagL0$  puede establecerse en 1, el segundo indicador de predicción de dirección  $predFlagL1$  puede establecerse en 0, y el primer vector de movimiento de subbloque

de dirección mvLO puede establecerse en (0, 0).

5 S1204: Realizar una compensación de movimiento en el subbloque del bloque a procesar en función del vector de movimiento del subbloque del bloque a procesar y del fotograma de referencia del bloque a procesar, para obtener un valor de predicción del subbloque del bloque a procesar.

10 Debe entenderse que, el procedimiento anterior de procesamiento del subbloque de ejemplo se realiza en cada subbloque del bloque a procesar, para obtener un valor de predicción de cada subbloque. El bloque a procesar incluye subbloques. Por lo tanto, después de que se determina el valor de predicción de cada subbloque, se determina en consecuencia un valor de predicción del bloque a procesar.

15 En esta realización, se proporcionan varios métodos simplificados para obtener el vector de desplazamiento temporal. Esto reduce la complejidad del cálculo al tiempo que garantiza la precisión de obtener el subbloque colocado. Además, se simplifica una manera de determinar el vector de movimiento del subbloque del subbloque correspondiente a procesar cuando el subbloque colocado no está disponible, y la complejidad del cálculo se reduce aún más.

20 Como se muestra en la FIG. 13, un bloque a procesar incluye uno o más subbloques. Un aparato de interpredicción 1300 incluye:

25 un módulo de obtención de desplazamiento 1301, configurado para determinar un vector de desplazamiento temporal del bloque a procesar basado en un bloque espacialmente vecino del bloque a procesar, donde el vector de desplazamiento temporal se utiliza para determinar un subbloque colocado del subbloque del bloque a procesar; y un vector de movimiento que obtiene el módulo 1302, configurado para determinar un vector de movimiento del subbloque al procesar basado en un vector de movimiento del subbloque colocado, donde el vector de movimiento del subbloque al procesar se obtiene basado en un primer vector de movimiento preestablecido cuando el vector de movimiento del subbloque colocado no está disponible.

30 En una implementación factible, el módulo de obtención de desplazamiento 1301 se configura específicamente para: comprobar secuencialmente, en un orden preestablecido, si los vectores de movimiento de bloques espacialmente vecinos en una pluralidad de primeras posiciones preestablecidas están disponibles, hasta que se obtiene un vector de movimiento de un primer bloque espacialmente vecino cuyo vector de movimiento está disponible en el orden preestablecido; y utilizar, como el vector de desplazamiento temporal, el vector de movimiento del primer bloque espacialmente vecino cuyo vector de movimiento está disponible en el orden preestablecido.

35 En una implementación factible, el módulo de obtención de desplazamiento 1301 se configura específicamente para utilizar un segundo vector de movimiento preestablecido como el vector de desplazamiento temporal cuando los vectores de movimiento de los bloques espacialmente vecinos en la pluralidad de primeras posiciones preestablecidas no están disponibles.

40 En una implementación factible, el segundo vector de movimiento preestablecido es un vector de movimiento cero.

45 En una implementación factible, el módulo de obtención de desplazamiento 1301 se configura específicamente para: obtener un vector de movimiento y un fotograma de referencia de un bloque espacialmente vecino en una segunda posición preestablecida, donde el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida está disponible; y utilizar el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida como el vector de desplazamiento temporal.

50 En una implementación factible, el módulo de obtención de desplazamiento 1301 se configura específicamente para utilizar un tercer vector de movimiento preestablecido como vector de desplazamiento temporal cuando el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida no están disponibles.

55 En una implementación factible, el tercer vector de movimiento preestablecido es un vector de movimiento cero.

60 En una implementación factible, el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida incluye un primer vector de movimiento de dirección que se basa en la primera lista de fotogramas de referencia, y el fotograma de referencia del bloque espacialmente vecino en la segunda posición preestablecida incluye un primer fotograma de referencia de dirección correspondiente al primer vector de movimiento de dirección; y el módulo de obtención de desplazamiento 1301 se configura específicamente para utilizar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es el mismo que un fotograma de imagen en el que se encuentra el subbloque colocado.

65

5 En una implementación factible, cuando el primer fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado, el módulo de obtención de desplazamiento 1301 se configura específicamente para usar el tercer vector de movimiento preestablecido como el vector de desplazamiento temporal.

10 En una implementación factible, cuando se utiliza la predicción bidireccional para el bloque espacialmente vecino en la segunda posición preestablecida, el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida incluye además un segundo vector de movimiento de dirección que se basa en la segunda lista de fotogramas de referencia, y el fotograma de referencia del bloque espacialmente vecino en la segunda posición preestablecida incluye un segundo fotograma de referencia de dirección correspondiente al segundo vector de movimiento de dirección; y cuando el primer fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra un bloque temporal correspondiente del bloque a procesar, el módulo de obtención de desplazamiento 1301 se configura específicamente para: usar el segundo vector de movimiento de dirección como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque o subbloque de uso de imagen, o usar el tercer vector de movimiento preestablecido como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado.

20 En una implementación factible, cuando se utiliza la predicción bidireccional para el bloque espacialmente vecino en la segunda posición preestablecida, el vector de movimiento del bloque espacialmente vecino en la segunda posición preestablecida incluye un primer vector de movimiento de dirección que se basa en la primera lista de fotogramas de referencia y un segundo vector de movimiento de dirección que se basa en la segunda lista de fotogramas de referencia, y el fotograma de referencia del bloque espacialmente vecino en la segunda posición preestablecida incluye un primer fotograma de referencia de dirección correspondiente al primer vector de movimiento de dirección y un segundo fotograma de referencia de dirección correspondiente al segundo vector de movimiento de dirección; y el módulo de obtención de desplazamiento 1301 se configura específicamente para: cuando se obtiene un fotograma de imagen en el que se encuentra el segundo subbloque de referencia de la lista de fotogramas de referencia, utilizar el segundo vector de movimiento de dirección como segundo vector de movimiento temporal cuando el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado, o utilizar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado y el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado; o cuando un fotograma de imagen en el que se encuentra el subbloque colocado se obtiene de la primera lista de fotogramas de referencia, utilizar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado, o utilizar el segundo vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado.

45 En una implementación factible, el módulo de obtención de desplazamiento 1301 se configura específicamente para: cuando el fotograma de imagen en el que se encuentra el subbloque colocado se obtiene de la segunda lista de fotogramas de referencia, y todos los fotogramas de referencia en una lista de fotogramas de referencia del bloque a procesar se muestran antes de un fotograma de imagen en el que se encuentra el bloque a procesar, usar el segundo vector de movimiento de dirección como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado, o usar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado; o cuando el fotograma de imagen en el que se encuentra el subbloque colocado se obtiene de la primera lista de fotogramas de referencia, o al menos un fotograma de referencia en una lista de fotogramas de referencia del bloque a procesar se muestra después de un fotograma de imagen en el que se encuentra el bloque a procesar, utilizar el primer vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado, o utilizar el segundo vector de movimiento de dirección como vector de desplazamiento temporal cuando el primer fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado y el segundo fotograma de referencia de dirección es el mismo que el fotograma de imagen en el que se encuentra el subbloque colocado.

65 En una implementación factible, el módulo de obtención de desplazamiento 1301 se configura específicamente para usar el tercer vector de movimiento preestablecido se utiliza como vector de desplazamiento temporal cuando el segundo fotograma de referencia de dirección es diferente del fotograma de imagen en el que se encuentra el subbloque colocado y el primer fotograma de referencia de dirección es diferente del fotograma

de imagen en el que se encuentra el subbloque colocado.

5 En una implementación factible, se obtiene un índice que es del fotograma de imagen en el que se encuentra el subbloque colocado y que está en una lista de fotogramas de referencia del bloque espacialmente vecino del bloque a procesar analizando un flujo de bits.

10 En una implementación factible, una condición de que el vector de movimiento del bloque espacialmente vecino no está disponible incluye uno o una combinación de los siguientes elementos: el bloque espacialmente vecino no se codifica/decodifica; se utiliza un modo de intrapredicción o un modo de copia intrabloque para el bloque espacialmente vecino; el bloque espacialmente vecino no existe; o el bloque espacialmente vecino y el bloque a procesar se encuentran en diferentes regiones de codificado.

15 En una implementación factible, la región de codificado incluye una imagen, un segmento, un mosaico o un grupo de mosaicos.

20 En una implementación factible, el aparato incluye además: un módulo de determinación 1303, configurado para determinar si está disponible un vector de movimiento correspondiente a una posición de intrabloque preestablecida del subbloque colocado; y, en consecuencia, el módulo de obtención de vector de movimiento 1302 se configura específicamente para: obtener el vector de movimiento del subbloque del bloque a procesar en función del vector de movimiento correspondiente a la posición de intrabloque preestablecida cuando el vector de movimiento correspondiente a la posición de intrabloque preestablecida está disponible, u obtener el vector de movimiento del subbloque del bloque a procesar en función del primer vector de movimiento preestablecido cuando el vector de movimiento correspondiente a la posición de intrabloque preestablecida no está disponible.

25 En una implementación factible, la posición intrabloque preestablecida es una posición central geométrica del subbloque colocado.

30 En una implementación factible, cuando el modo de intrapredicción o el modo de copia intrabloque se utilizan para una unidad de predicción en la que se encuentra la posición intrabloque preestablecida, el vector de movimiento correspondiente a la posición intrabloque preestablecida no está disponible; o cuando la interpredicción se utiliza para una unidad de predicción en la que se encuentra la posición intrabloque preestablecida, el vector de movimiento correspondiente a la posición intrabloque preestablecida está disponible.

35 En una implementación factible, el módulo de obtención de vectores de movimiento 1302 se configura específicamente para usar el primer vector de movimiento preestablecido como el vector de movimiento del subbloque del bloque a procesar.

40 En una implementación factible, el primer vector de movimiento preestablecido es un vector de movimiento cero.

45 En una implementación factible, el vector de movimiento del subbloque incluye un primer vector de movimiento de subbloque de dirección que se basa en la primera lista de fotogramas de referencia y/o un segundo vector de movimiento de subbloque de dirección que se basa en la segunda lista de fotogramas de referencia; y cuando el vector de movimiento correspondiente a la posición de intrabloque preestablecida no está disponible, el módulo de obtención de vector de movimiento 1302 se configura específicamente para: determinar que la predicción unidireccional basada en el primer vector de movimiento de subbloque de dirección se utiliza para el subbloque del bloque a procesar, y obtener el primer vector de movimiento de subbloque de dirección del subbloque del subbloque a procesar en función del primer vector de movimiento preestablecido; o determinar que la predicción unidireccional basada en el segundo vector de movimiento de subbloque de dirección se utiliza para el subbloque del bloque a procesar y obtener el segundo vector de movimiento de subbloque de dirección del subbloque del bloque a procesar en función del primer vector de movimiento preestablecido.

55 En una implementación factible, cuando el vector de movimiento correspondiente a la posición de intrabloque preestablecida no está disponible, el módulo de obtención de vector de movimiento 1302 se configura específicamente para: cuando un tipo de predicción de una región de codificado en la que se encuentra el bloque a procesar es predicción tipo B, determinar que la predicción bidireccional se utiliza para el subbloque del bloque a procesar, y obtener por separado el primer vector de movimiento de subbloque de dirección del subbloque del bloque a procesar y el segundo vector de movimiento de subbloque de dirección del subbloque del bloque a procesar basado en el primer vector de movimiento preestablecido; o cuando un tipo de predicción de una región de codificado en la que se encuentra el bloque a procesar es predicción tipo P, determinar que la predicción direccional se utiliza para el subbloque del bloque a procesar y para obtener el primer vector de movimiento de subbloque de dirección del subbloque del bloque a procesar en función del primer vector de movimiento preestablecido.

5 En una implementación factible, el módulo de obtención de vector de movimiento 1302 se configura específicamente para realizar, en función de una relación de una primera diferencia de distancia temporal a una segunda diferencia de distancia temporal, procesar con escalado en el vector de movimiento correspondiente a la posición preestablecida intrabloque, para obtener el vector de movimiento del subbloque del bloque a procesar, donde la primera diferencia de distancia temporal es una diferencia de recuento de orden de imagen entre el fotograma de imagen en el que se encuentra el bloque a procesar y un fotograma de referencia del bloque a procesar, y la segunda diferencia de distancia temporal es una diferencia de recuento de orden de imagen entre el fotograma de imagen en el que se encuentra el subbloque colocado y un fotograma de referencia del subbloque colocado.

10 En una implementación factible, se obtiene un índice del fotograma de referencia del bloque a procesar en la lista de fotogramas de referencia del bloque a procesar analizando el flujo de bits.

15 En una implementación factible, se obtiene el índice del fotograma de referencia del bloque a procesar en la lista de fotogramas de referencia del bloque a procesar es 0.

20 En una implementación factible, el aparato incluye además: un módulo de compensación de movimiento 1304, configurado para realizar la compensación de movimiento en el subbloque del bloque a procesar en función del vector de movimiento del subbloque del bloque a procesar y el fotograma de referencia del bloque a procesar, para obtener un valor de predicción del subbloque del bloque a procesar.

25 Debe entenderse que los módulos en esta realización de esta solicitud que se muestran en la FIG. 13 se configuran para realizar el método que se muestra en la FIG. 12 y las implementaciones factibles, y tienen un mismo efecto técnico.

30 A continuación se describen, mediante ejemplos, algunas implementaciones relacionadas con las realizaciones de esta solicitud.

Realización A

35 Etapa 1: Determinar un vector de movimiento de desplazamiento.

40 El vector de movimiento de desplazamiento se utiliza para determinar la posición de un bloque colocado de una CU actual en una imagen colocada. Un vector de movimiento de un bloque espacialmente vecino de la CU actual se puede utilizar como vector de movimiento de desplazamiento. Una posición de la CU actual en la imagen colocada se determina en función del vector de movimiento de desplazamiento y una imagen colocada de un fotograma actual. Un bloque en la posición se denomina bloque colocado (bloque correspondiente/colocado). El vector de movimiento de desplazamiento puede obtenerse utilizando uno de los siguientes métodos.

Método 1: En una implementación, como se muestra en la FIG. 6, si A1 está disponible, el vector de movimiento de desplazamiento se determina según el siguiente método.

45 Si A1 no está disponible, el valor del vector de movimiento es 0. Que A1 no esté disponible significa que si un bloque en la posición A1 no está decodificado, el bloque no es un bloque de interpredicción (es un bloque de intrapredicción o un bloque de copia intrabloque (IBC)), o si el bloque se encuentra fuera de un segmento, mosaico, grupo de mosaicos o imagen actual, se considera que el bloque es un bloque no disponible.

50 Si se cumplen todas las condiciones siguientes, el vector de movimiento de desplazamiento del bloque actual es un vector de movimiento correspondiente a una lista 1 de A1.

La lista 1 se utiliza para A1 para la predicción.

55 Un fotograma de referencia en la lista 1 de A1 utilizado para la predicción es el mismo que la imagen colocada del fotograma actual. (Se determina si los POC son iguales, un índice del fotograma de referencia y un índice de la imagen colocada corresponden a un mismo POC, y el índice de la imagen colocada del bloque actual se puede obtener del flujo de bits.)

60 Se utiliza una estructura de codificado de baja latencia, para ser específicos, solo una imagen que se muestra antes de que se utilice una imagen actual para la predicción.

Un tipo de una imagen, un mosaico o un grupo de mosaicos en el que se encuentra el bloque actual es un tipo B.

65 `collocated_from_10_flag` es 0. `collocated_from_10_flag` igual a 1 especifica que una imagen colocada para la predicción de vector de movimiento temporal se obtiene de una lista de colas de imágenes de referencia 0.

collocated\_from\_10\_flag igual a 0 especifica que una imagen colocada para la predicción de vector de movimiento temporal se obtiene de una lista de colas de imágenes de referencia 1. Un valor de collocated\_from\_10\_flag es 1 cuando collocated\_from\_10\_flag no está presente en el flujo de bits.

- 5 De lo contrario, si se cumplen todas las condiciones siguientes, el vector de movimiento de desplazamiento es un vector de movimiento correspondiente a una lista 0 de A1.

La lista 0 se utiliza para A1 para la predicción.

- 10 Un fotograma de referencia en la lista 0 de A1 utilizado para la predicción es el mismo que la imagen colocada del fotograma actual.

Método 2: Por ejemplo, un vector de movimiento de un primer bloque vecino disponible se encuentra en una secuencia de A1, B1, B0 y A0 en la FIG. 6. Si el vector de movimiento apunta a la imagen colocada, el vector de movimiento se utiliza como vector de movimiento de desplazamiento de la CU actual. De lo contrario, se puede usar un vector de movimiento cero; o el vector de movimiento del primer bloque vecino disponible se escala, de modo que el vector de movimiento cero apunta a la imagen colocada, y se usa un vector de movimiento escalado como el vector de movimiento de desplazamiento de la CU actual.

- 20 Debe entenderse que el vector de desplazamiento utilizado puede ser alternativamente un vector de desplazamiento cero. En este caso, un bloque de imagen en la imagen colocada que está en la misma posición que el bloque a procesar es un bloque colocado del bloque a procesar en la imagen colocada. Además, cuando no se puede encontrar un vector de desplazamiento de destino que cumpla con un requisito, no se puede usar una tecnología ATMVP, sino que se usa otra tecnología para obtener un vector de movimiento de un subbloque a procesar.

Etapa 2: Obtener información de disponibilidad e información de movimiento predeterminada que están en modo ATMVP.

- 30 El bloque colocado puede obtenerse primero en función del vector de desplazamiento, y se obtiene un modo de predicción de un subbloque colocado S en el que se ubica una posición preestablecida en el bloque colocado. Las coordenadas (xcol, ycol) de la posición preestablecida del bloque colocado se pueden obtener según una fórmula (6). La información de movimiento predeterminada y la disponibilidad en el modo ATMVP se obtienen en función del modo de predicción y la información de movimiento del subbloque S colocado. Un método específico es el siguiente.

Para ser específicos, las coordenadas de la posición preestablecida se determinan primero en el bloque colocado, para obtener el modo de predicción del subbloque correspondiente a la posición preestablecida se encuentra; y si el ATMVP está actualmente disponible se determina en función del modo de predicción del subbloque S correspondiente a la posición preestablecida se encuentra.

$$\begin{cases} x_{col} = x + \frac{W}{2} + x_{off} \\ y_{col} = y + \frac{H}{2} + y_{off} \end{cases} \quad (6)$$

- 45 (x, y) representa las coordenadas de una esquina superior izquierda de la CU actual, W representa la anchura de la CU actual, H representa la altura de la CU actual y (x<sub>off</sub>, y<sub>off</sub>) representa el vector de movimiento de desplazamiento.

- 50 Si el modo de predicción del subbloque S colocado en el que se encuentra la posición preestablecida es un modo de intrapredicción o un modo de copia dentro del bloque, la información de movimiento en el modo ATMVP no está disponible.

- 55 Si el modo de predicción del subbloque S colocado en el que se encuentra la posición preestablecida es un modo de interpredicción, la información de movimiento del subbloque S colocado se extrae adicionalmente, y las coordenadas de la posición preestablecida se obtienen según la fórmula (6). Además, la información de movimiento de la posición en un campo de vector de movimiento de la imagen colocada se utiliza como la información de movimiento del subbloque S colocada, donde la información de movimiento del subbloque S colocada se conoce como información de movimiento predeterminada del subbloque S colocada.

- 60 Se escala un vector de movimiento por defecto MV del subbloque S colocado, para obtener un vector de movimiento por defecto (MV) del subbloque a procesar, y se utiliza un vector de movimiento escalado (MV) como información de movimiento por defecto.

Por ejemplo, como se muestra en la FIG. 10, el MV escalado se puede obtener utilizando un método en una

fórmula (7). Un método de escalado no se limita específicamente en la presente memoria.

$$MV_s = \frac{CurPoc - CurRefPoc}{ColPoc - ColRefPoc} \times MV \quad (7)$$

- 5 Se supone que un POC de un fotograma en el que se encuentra el bloque actual es CurPoc, un POC de un fotograma de referencia del bloque actual es CurRefPoc, un POC de la imagen colocada es ColPoc, un POC de un fotograma de referencia del subbloque colocado es ColRefPoc y un vector de movimiento a escala futura es *MV*.
- 10 Opcionalmente, el *MV* se descompone en un vector de movimiento *MVx* en una dirección horizontal y un vector de movimiento *MVy* en una dirección vertical, y un vector de movimiento *MVsx* en la dirección horizontal y un vector de movimiento *MVsy* en la dirección vertical se obtienen por separado a través del cálculo según la fórmula anterior.
- 15 (Debe entenderse que la posición preestablecida corresponde a un solo subbloque *S*, y la selección del subbloque *S* asignado es para determinar, en función del modo de predicción del subbloque *S* asignado, si el ATMVP está disponible.)

20 Etapa 3: Determinar información de movimiento del subbloque a procesar en función de la información de movimiento del subbloque colocado.

Como se muestra en la FIG. 11, para cada subbloque en la CU actual, se determina un subbloque colocado del subbloque en la imagen colocada en función del vector de movimiento de desplazamiento y las coordenadas de posición del subbloque, y se obtiene información de movimiento del subbloque colocado.

- 25 Las coordenadas de posición de un punto central del subbloque colocado se obtienen según una fórmula (8). (*x*, *y*) representa las coordenadas de una esquina superior izquierda de la actual CU, *i* representa un *i*-ésimo subbloque de izquierda a derecha, *j* representa un *j*-ésimo subbloque de arriba a abajo, (*x*<sub>off</sub>, *y*<sub>off</sub>) representa el vector de movimiento de desplazamiento, *M*×*N* representa un tamaño del subbloque (por ejemplo, 4 x 4 u 8 x 8), y (*x*<sub>(*i*,*j*)</sub>, *y*<sub>(*i*,*j*)</sub>) representa las coordenadas de posición de un correspondiente subbloque (*i*, *j*)-ésimo.
- 30

$$\begin{cases} x_{(i,j)} = x + M \times i + \frac{M}{2} + x_{off} \\ y_{(i,j)} = y + N \times j + \frac{N}{2} + y_{off} \end{cases} \quad (8)$$

- 35 Se obtiene un modo de predicción del subbloque colocado basado en las coordenadas de posición del punto central del subbloque colocado. Si el modo de predicción del subbloque colocado es el modo de interpretación, la información de movimiento del subbloque colocado está disponible. En este caso, la información de movimiento de la posición en el campo de vector de movimiento de la imagen colocada se utiliza como información de movimiento del subbloque colocado. La información de movimiento del subbloque colocado se escala, para obtener la información de movimiento del subbloque a procesar. Un método de escalado es el mismo que el método en la etapa 2, y los detalles no se describen de nuevo en esta memoria.
- 40

- 45 Si el modo de predicción del subbloque colocado es el modo de intrapredicción o el modo de copia intrabloque, la información de movimiento del subbloque colocado no está disponible. En este caso, la información de movimiento predeterminada obtenida en la etapa 2 se puede utilizar como la información de movimiento del subbloque colocado.

Etapa 4: Realizar una predicción de compensación de movimiento basada en la información de movimiento de cada subbloque, para obtener un valor de píxel predicho de la CU actual.

- 50 Se añade un vector de movimiento a las coordenadas de un píxel en una esquina superior izquierda de cada subbloque en función de la información de movimiento del subbloque, para encontrar un punto de coordenadas correspondiente en un fotograma de referencia. Si el vector de movimiento tiene una precisión de muestra fraccionada, es necesario realizar un filtrado de interpolación para obtener un valor de píxel previsto del subbloque. De lo contrario, un valor de píxel en el fotograma de referencia se obtiene directamente y se utiliza como un valor de píxel previsto del subbloque.
- 55

- Es necesario introducir el vector de movimiento de desplazamiento para determinar la información de disponibilidad que se encuentra en el modo ATMVP. Esto depende de un proceso de búsqueda de vector de movimiento de desplazamiento. Además, cuando la información de movimiento del subbloque colocado no está disponible, es necesario calcular la información de movimiento predeterminada. Esto afecta a la velocidad de codificado.
- 60

Realización B

Etapa 1: Determinar un vector de movimiento de desplazamiento.

5 El vector de movimiento de desplazamiento se utiliza para determinar la posición de un bloque colocado de una CU actual en una imagen colocada. Un vector de movimiento de un bloque espacialmente vecino de la CU actual se puede utilizar como vector de movimiento de desplazamiento. Se utiliza un fotograma de referencia del bloque espacialmente vecino como la imagen colocada de la CU actual. Una posición de la CU actual en un bloque de imagen colocada se determina en función del vector de movimiento de desplazamiento y el bloque de imagen colocada. Un bloque en la posición se denomina bloque colocado (bloque correspondiente/colocado).

Un método específico es el mismo que en la etapa 1 en la realización A.

Etapa 2: Obtener información de movimiento de un subbloque colocado.

15 Como se muestra en la FIG. 11, el bloque colocado se puede obtener primero en función del vector de desplazamiento, y luego un subbloque colocado que tiene una relación de posición relativa con un subbloque a procesar se determina en una imagen objetivo en función de una posición del subbloque a procesar (también se puede entender como que un subbloque colocado que tiene una relación de posición relativa con el subbloque a procesar se determina en el bloque colocado).

Para cada subbloque en la CU actual, se determina un subbloque colocado del subbloque en la imagen colocada en función del vector de movimiento de desplazamiento y las coordenadas de posición del subbloque, y se obtiene información de movimiento del subbloque colocado.

25 Las coordenadas de posición de un punto central del subbloque colocado se obtienen según una fórmula (9).  $(x, y)$  representa las coordenadas de una esquina superior izquierda de la actual CU,  $i$  representa un  $i$ -ésimo subbloque de izquierda a derecha,  $j$  representa un  $j$ -ésimo subbloque de arriba a abajo,  $(x_{off}, y_{off})$  representa el vector de movimiento de desplazamiento,  $M \times N$  representa un tamaño del subbloque (por ejemplo,  $4 \times 4$  u  $8 \times 8$ ), y  $(x_{(i,j)}, y_{(i,j)})$  representa coordenadas de posición de un correspondiente subbloque  $(i, j)$ -ésimo.

$$\begin{cases} x_{(i,j)} = x + M \times i + \frac{M}{2} + x_{off} \\ y_{(i,j)} = y + N \times j + \frac{N}{2} + y_{off} \end{cases} \quad (9)$$

35 Se obtiene un modo de predicción del subbloque colocado en función de las coordenadas de posición del punto central del subbloque colocado. Si el modo de predicción del subbloque colocado es un modo de interpredicción, la información de movimiento del subbloque colocado está disponible. En este caso, la información de movimiento de la posición en un campo de vector de movimiento de la imagen colocada se utiliza como información de movimiento del subbloque colocado. La información de movimiento del subbloque actual se deriva en función de la información de movimiento del subbloque colocado. Se escala un vector de movimiento del subbloque colocado y se convierte en un vector de movimiento del subbloque. Se puede utilizar un método de escalado en la tecnología convencional, y no se describen detalles en esta memoria.

45 Si el modo de predicción del subbloque colocado es un modo de intrapredicción o un modo de copia intrabloque, la información de movimiento del subbloque colocado no está disponible. En este caso, se puede utilizar uno de los siguientes métodos de procesamiento.

Método 1: Si un tipo de una imagen, un segmento o un grupo de mosaicos en el que se encuentra el subbloque de la CU actual es un tipo B, se rellena un vector de movimiento cero bidireccional en el subbloque coloquial o el subbloque a procesar, por ejemplo,  $\text{predFlagL0} = 1$ ,  $\text{predFlagL1} = 1$ ,  $\text{mvL0} = 0$  y  $\text{mvL1} = 0$ .

50 De lo contrario, se rellena un vector de movimiento de lista unidireccional 0, por ejemplo,  $\text{predFlagL0} = 1$ ,  $\text{predFlagL1} = 0$ ,  $\text{mvL0} = 0$  y  $\text{mvL1} = 0$ .

Método 2: Información del vector de movimiento cero de lista unidireccional 0 se rellena en el subbloque colocado o el subbloque a procesar, por ejemplo,  $\text{predFlagL0} = 1$ ,  $\text{predFlagL1} = 0$ ,  $\text{mvL0} = 0$  y  $\text{mvL1} = 0$ .

Método 3: información del vector de movimiento cero de lista unidireccional 1 se rellena en el subbloque colocado o el subbloque a procesar, por ejemplo,  $\text{predFlagL0} = 0$ ,  $\text{predFlagL1} = 1$ ,  $\text{mvL0} = 0$  y  $\text{mvL1} = 0$ .

60  $\text{predFlagL0}$  y  $\text{predFlagL1}$  respectivamente representan direcciones de predicción para realizar predicción mediante el uso de una lista 0 y una lista 1, y  $\text{mvL0}$  y  $\text{mvL1}$  respectivamente representan vectores de movimiento utilizados para realizar la predicción mediante el uso de la lista 0 y la lista 1.  $\text{mvL0} = 0$  representa que tanto un componente horizontal como un componente vertical de  $\text{mvL0}$  están llenos de 0, y  $\text{mvL1} = 0$

representa que tanto un componente horizontal como un componente vertical de mvL1 son 0.

Etapa 3: Realizar una predicción de compensación de movimiento basada en la información de movimiento de cada subbloque, para obtener un valor de píxel predicho de la CU actual.

5

Se añade un vector de movimiento a las coordenadas de un píxel en una esquina superior izquierda de cada subbloque en función de la información de movimiento del subbloque, para encontrar un punto de coordenadas correspondiente en el fotograma de referencia. Si el vector de movimiento tiene una precisión de muestra fraccionada, es necesario realizar un filtrado de interpolación para obtener un valor de píxel previsto del subbloque. De lo contrario, un valor de píxel en el fotograma de referencia se obtiene directamente y se utiliza como un valor de píxel previsto del subbloque.

10

Las enmiendas al documento son las siguientes (para una base de las enmiendas, consúltese JVET-N1001-v3, para el significado del siguiente pseudocódigo, consúltese este documento, y este documento se puede descargar del sitio web <http://phenix.int-evry.fr/jvet/>):

15

– Cuando availableFlagL0SbCo1 y availableFlagL1SbCo1 son iguales a 0, se aplica lo siguiente:

$$mvL0SbCol[xSbIdx][ySbIdx][0] = 0 \quad (8-638)$$

$$mvL0SbCol[xSbIdx][ySbIdx][1] = 0 \quad (8-638)$$

$$predFlagL0SbCol[xSbIdx][ySbIdx] = 1 \quad (8-639)$$

$$mvL1SbCol[xSbIdx][ySbIdx][0] = 0 \quad (8-638)$$

$$mvL1SbCol[xSbIdx][ySbIdx][1] = 0 \quad (8-638)$$

$$predFlagL1SbCol[xSbIdx][ySbIdx] = slice\_type == B ? 1 : 0 \quad (8-639)$$

$$mvL0SbCol[xSbIdx][ySbIdx][0] = 0 \quad (8-638)$$

$$mvL0SbCol[xSbIdx][ySbIdx][1] = 0 \quad (8-638)$$

$$predFlagL0SbCol[xSbIdx][ySbIdx] = 1 \quad (8-639)$$

$$mvL1SbCol[xSbIdx][ySbIdx][0] = 0 \quad (8-638)$$

$$mvL1SbCol[xSbIdx][ySbIdx][1] = 0 \quad (8-638)$$

$$predFlagL1SbCol[xSbIdx][ySbIdx] = 0 \quad (8-639)$$

20

Según esta realización de esta solicitud, cuando la información de movimiento del subbloque colocado no está disponible, se resuelve un problema de tecnología convencional que requiere el cálculo complejo de vector de movimiento de desplazamiento de inicio, y la información de movimiento preestablecida se llena directamente, para determinar la información de disponibilidad y la información de movimiento predeterminada que están en el modo ATMVP. Esto reduce la complejidad del codificado.

25

#### Realización C

Esta realización se refiere a un método de interpredicción, y se optimiza un método para obtener un vector de movimiento de desplazamiento. Las etapas 2 y 3 son las mismas que los de la realización A. A continuación se ofrece una descripción específica.

30

Etapa 1: Determinar un vector de movimiento de desplazamiento.

35

El vector de movimiento de desplazamiento se utiliza para determinar la posición de un bloque colocado de

una CU actual en una imagen colocada. Un vector de movimiento de un bloque espacialmente vecino de la CU actual se puede utilizar como vector de movimiento de desplazamiento. Se utiliza un fotograma de referencia del bloque espacialmente vecino como la imagen colocada de la CU actual. Una posición de la CU actual en un bloque de imagen colocada se determina en función del vector de movimiento de desplazamiento y el bloque de imagen colocada. Un bloque en la posición se denomina bloque colocado (bloque correspondiente/colocado).

Como se muestra en la FIG. 10, si A1 está disponible, y una imagen colocada a la que un vector de movimiento de puntos A1 es una imagen colocada de A1 (es decir, una imagen de referencia de A1 es un fotograma colocada de A1), el vector de movimiento de A1 se utiliza como vector de movimiento desplazado de la CU actual. El vector de movimiento de desplazamiento puede obtenerse utilizando uno de los siguientes métodos. Si A1 no está disponible, el valor del vector de movimiento es 0. Que A1 no esté disponible significa que si un bloque en la posición A1 no está decodificado, el bloque no es un bloque de intrapredicción o un bloque de copia intrabloque, o si el bloque se encuentra fuera de un segmento, mosaico, grupo de mosaicos o imagen actual, se considera que el bloque es un bloque no disponible.

Método 1: Se determina si se cumplen todas las siguientes condiciones preestablecidas. Si se cumplen todas las condiciones preestablecidas, se comprueba si un fotograma de referencia correspondiente a una lista 1 de A1 es el mismo que una imagen colocada de un fotograma actual. Si el fotograma de referencia correspondiente a la lista 1 de A1 es el mismo que la imagen colocada del fotograma actual, se utiliza un vector de movimiento correspondiente a la lista 1 como vector de movimiento de desplazamiento; o si el fotograma de referencia correspondiente a la lista 1 de A1 es diferente de la imagen colocada del fotograma actual, se comprueba si un fotograma de referencia correspondiente a una lista 0 es el mismo que la imagen colocada del fotograma actual. Si el fotograma de referencia correspondiente a la lista 0 es el mismo que la imagen colocada del fotograma actual, se utiliza un vector de movimiento correspondiente a la lista 0 como vector de movimiento de desplazamiento; de lo contrario, el vector de movimiento de desplazamiento es 0.

(1) Se utiliza una estructura de codificado de baja latencia, para ser específicos, solo una imagen que se muestra antes de que se utilice una imagen actual para la predicción.

(2) Un tipo de una imagen, un mosaico o un grupo de mosaicos en el que se encuentra el bloque actual es un tipo B.

(3) `collocated_from_10_flag` es 0. `collocated_from_10_flag` igual a 1 especifica que una imagen colocada para la predicción de vector de movimiento temporal se obtiene de una lista de colas de imágenes de referencia 0. `collocated_from_10_flag` igual a 0 especifica que una imagen colocada para la predicción de vector de movimiento temporal se obtiene de una lista de colas de imágenes de referencia 1. Un valor de `collocated_from_10_flag` es 1 cuando `collocated_from_10_flag` no está presente en el flujo de bits.

De lo contrario (si no se cumplen las condiciones preestablecidas anteriores), se comprueba primero si el fotograma de referencia correspondiente a la lista 0 de A1 es el mismo que la imagen colocada del fotograma actual. Si el fotograma de referencia correspondiente a la lista 0 de A1 es el mismo que la imagen colocada del fotograma actual, se utiliza el vector de movimiento correspondiente a la lista 0 como vector de movimiento de desplazamiento; o si el fotograma de referencia correspondiente a la lista 0 de A1 es diferente de la imagen colocada del fotograma actual, se comprueba si el fotograma de referencia correspondiente a la lista 1 es el mismo que la imagen colocada del fotograma actual. Si el fotograma de referencia correspondiente a la lista 1 es el mismo que la imagen colocada del fotograma actual, se utiliza el vector de movimiento correspondiente a la lista 1 como vector de movimiento de desplazamiento; de lo contrario, el vector de movimiento de desplazamiento es 0.

Método 2: Se determina si se cumplen ambas de las siguientes condiciones preestablecidas. Si se cumplen ambas de las condiciones preestablecidas, se comprueba si un fotograma de referencia correspondiente a una lista 1 de A1 es el mismo que una imagen colocada de un fotograma actual. Si el fotograma de referencia correspondiente a la lista 1 de A1 es el mismo que la imagen colocada del fotograma actual, se utiliza un vector de movimiento correspondiente a la lista 1 como vector de movimiento de desplazamiento; o si el fotograma de referencia correspondiente a la lista 1 de A1 es diferente de la imagen colocada del fotograma actual, se comprueba si un fotograma de referencia correspondiente a una lista 0 es el mismo que la imagen colocada del fotograma actual. Si el fotograma de referencia correspondiente a la lista 0 es el mismo que la imagen colocada del fotograma actual, se utiliza un vector de movimiento correspondiente a la lista 0 como vector de movimiento de desplazamiento; de lo contrario, el vector de movimiento de desplazamiento es 0.

(1) Un tipo de una imagen, un mosaico o un grupo de mosaicos en el que se encuentra el bloque actual es un tipo B.

(2) `collocated_from_10_flag` es 0. `collocated_from_10_flag` igual a 1 especifica que una imagen colocada para la predicción de vector de movimiento temporal se obtiene de una lista de colas de imágenes de referencia 0.

collocated\_from\_10\_flag igual a 0 especifica que una imagen colocada para la predicción de vector de movimiento temporal se obtiene de una lista de colas de imágenes de referencia 1. Un valor de collocated\_from\_10\_flag es 1 cuando collocated\_from\_10\_flag no está presente en el flujo de bits.

5 De lo contrario (si no se cumplen las condiciones preestablecidas anteriores), se comprueba primero si el fotograma de referencia correspondiente a la lista 0 de A1 es el mismo que la imagen colocada del fotograma actual. Si el fotograma de referencia correspondiente a la lista 0 de A1 es el mismo que la imagen colocada del fotograma actual, se utiliza el vector de movimiento correspondiente a la lista 0 como vector de movimiento de desplazamiento; o si el fotograma de referencia correspondiente a la lista 0 de A1 es diferente de la imagen colocada del fotograma actual, se comprueba si el fotograma de referencia correspondiente a la lista 1 es el mismo que la imagen colocada del fotograma actual. Si el fotograma de referencia correspondiente a la lista 1 es el mismo que la imagen colocada del fotograma actual, se utiliza el vector de movimiento correspondiente a la lista 1 como vector de movimiento de desplazamiento; de lo contrario, el vector de movimiento de desplazamiento es 0.

15 Método 3: Primero se comprueba si un fotograma de referencia correspondiente a una lista 0 de A1 es el mismo que una imagen colocada de un fotograma actual. Si el fotograma de referencia correspondiente a la lista 0 de A1 es el mismo que la imagen colocada del fotograma actual, se utiliza un vector de movimiento correspondiente a la lista 0 de A1 como vector de movimiento de desplazamiento. No es necesario comprobar si un fotograma de referencia correspondiente a una lista 1 es el mismo que la imagen colocada del fotograma actual. Si el fotograma de referencia correspondiente a la lista 0 de A1 es diferente de la imagen colocada del fotograma actual, y un tipo de una imagen, un mosaico o un grupo de mosaicos en el que se encuentra el bloque actual es un tipo B, debe determinarse si el fotograma de referencia de la lista 1 es el mismo que la imagen colocada del fotograma actual. Si el fotograma de referencia de la lista 1 es el mismo que la imagen colocada del fotograma actual, puede utilizarse el vector de movimiento correspondiente a la lista 1 de A1 como vector de movimiento de desplazamiento; de lo contrario, el vector de movimiento de desplazamiento es 0.

30 Método 4: Se comprueba si un fotograma de referencia correspondiente a una lista 0 de A1 es el mismo que una imagen colocada de un fotograma actual. Si el fotograma de referencia correspondiente a la lista 0 de A1 es el mismo que la imagen colocada del fotograma actual, se utiliza un vector de movimiento correspondiente a la lista 0 de A1 como vector de movimiento de desplazamiento; de lo contrario, el vector de movimiento de desplazamiento es 0.

35 Se puede obtener un número de índice (idx) de la imagen colocada del fotograma actual del bloque de imagen del flujo de bits.

40 Debe entenderse que el vector de desplazamiento utilizado puede ser alternativamente un vector de desplazamiento cero. En este caso, un bloque de imagen en la imagen colocada que está en la misma posición que el bloque a procesar es un bloque colocado del bloque a procesar en la imagen colocada. Además, cuando no se puede encontrar un vector de desplazamiento de destino que cumpla con un requisito, no se puede usar una tecnología ATMVP, sino que se usa otra tecnología para obtener un vector de movimiento de un subbloque a procesar.

45 Modificaciones al documento son de la siguiente manera:

Método 1:

Cuando availableFlagA<sub>1</sub> es igual a TRUE, se aplica lo siguiente:

50 – Si son verdaderas todas las condiciones siguientes, checkL1First se establece igual a 1:  
 – DiffPicOrderCnt(aPic, currPic) es menor o igual a 0 para cada imagen aPic en cada lista de imágenes de referencia del segmento actual,

55 slice\_type es igual a B,

– collocated\_from\_10\_flag es igual a 0.

60 – K se establece igual a checkL1First, y si son verdaderas todas las condiciones siguientes, tempMV se establece igual a mvLKA<sub>1</sub>:

– predFlagLKA<sub>1</sub> es igual a 1,

– DiffPicOrderCnt(ColPic, RefPicList[ K ][ refIdxLKA<sub>1</sub>]) es igual a 0.

65 – De lo contrario, K se establece igual a ( 1 - checkL1First, ) y si son verdaderas todas las condiciones

siguientes, tempMV se establece igual a mvLKA<sub>1</sub>:

- predFlagLKA<sub>1</sub> es igual a 1,
- 5 – DiffPicOrderCnt(ColPic, RefPicList[ K ][ reldxLKA<sub>1</sub>]) es igual a 0.

Método 2:

Cuando availableFlagA<sub>1</sub> es igual a TRUE, se aplica lo siguiente:

- 10 – Si son verdaderas todas las condiciones siguientes, checkL1First se establece igual a 1:
- slice\_type es igual a B,
- 15 – collocated\_from\_10\_flag es igual a 0.
- K se establece igual a checkL1First, y si son verdaderas todas las condiciones siguientes, tempMV se establece igual a mvLKA<sub>1</sub>:
- 20 – predFlagLKA<sub>1</sub> es igual a 1,
- DiffPicOrderCnt(ColPic, RefPicList[ K ][ reldxLKA<sub>1</sub>]) es igual a 0.
- De lo contrario, K se establece igual a ( 1 - checkL1First, ) y si son verdaderas todas las condiciones siguientes, tempMV se establece igual a mvLKA<sub>1</sub>:
- 25 – predFlagLKA<sub>1</sub> es igual a 1,
- DiffPicOrderCnt(ColPic, RefPicList[ K ][ reldxLKA<sub>1</sub>]) es igual a 0.
- 30

Método 3:

Cuando availableFlagA<sub>1</sub> es igual a TRUE, se aplica lo siguiente:

- 35 – Si son verdaderas todas las condiciones siguientes, tempMV se establece igual a mvL0A<sub>1</sub>:
- predFlagL0A<sub>1</sub> es igual a 1,
- DiffPicOrderCnt(ColPic, RefPicList[ 0 ][ reldxL0A<sub>1</sub>]) es igual a 0.
- 40 – De lo contrario, si son verdaderas todas las condiciones siguientes, tempMV se establece igual a mvL1A<sub>1</sub>:
- slice\_type es igual a B,
- 45 – predFlagL1A<sub>1</sub> es igual a 1,
- DiffPicOrderCnt(ColPic, RefPicList[ 1 ][ reldxL1A<sub>1</sub>]) es igual a 0.

50 Método 4:

Cuando availableFlagA<sub>1</sub> es igual a TRUE, se aplica lo siguiente:

- 55 – Si son verdaderas todas las condiciones siguientes, tempMV se establece igual a mvL0A<sub>1</sub>:
- predFlagL0A<sub>1</sub> es igual a 1,
- DiffPicOrderCnt(ColPic, RefPicList[ 0 ][ reldxL0A<sub>1</sub>]) es igual a 0.

60 Realizaciones de esta solicitud proporcionan un nuevo método de cálculo de vector de movimiento de desplazamiento, para reducir la complejidad de codificado y aumentar la eficiencia de la codificado.

65 Un experto en la técnica puede entender que, las funciones descritas con referencia a diversos bloques lógicos ilustrativos, módulos y etapas de algoritmo divulgados y descritos en esta memoria descriptiva se pueden implementar mediante el uso de hardware, software, firmware o cualquier combinación de los mismos. Si se utiliza software para la implementación, las funciones descritas con referencia a los bloques lógicos ilustrativos,

módulos y etapas pueden almacenarse o transmitirse a través de un medio legible por ordenador como una o más instrucciones o código y ejecutarse por una unidad de procesamiento basada en hardware. El medio legible por ordenador puede incluir un soporte de almacenamiento legible por ordenador. El soporte de almacenamiento legible por ordenador corresponde a un medio tangible tal como un soporte de  
 5 almacenamiento de datos, o un medio de comunicaciones que incluye cualquier medio que facilita la transferencia de un programa informático de un lugar a otro lugar (por ejemplo, según un protocolo de comunicaciones). De esta manera, el medio legible por ordenador puede corresponder generalmente a: (1) un soporte de almacenamiento tangible no transitorio legible por ordenador o (2) un medio de comunicaciones tal como una señal o una portadora. El soporte de almacenamiento de datos puede ser cualquier medio utilizable  
 10 al que puedan acceder uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para implementar las tecnologías descritas en esta solicitud. Un producto de programa informático puede incluir el medio legible por ordenador.

A modo de ejemplo pero no limitación, tal soporte de almacenamiento legible por ordenador puede incluir una  
 15 RAM, una ROM, una EEPROM, un CD-ROM u otro aparato de almacenamiento de disco óptico, un aparato de almacenamiento de disco magnético u otro aparato de almacenamiento magnético, una memoria flash, o cualquier otro medio que se pueda configurar para almacenar código de programa deseado en forma de una instrucción o una estructura de datos y al que pueda acceder un ordenador. Además, cualquier conexión se puede denominar correctamente medio legible por ordenador. Por ejemplo, si se transmiten instrucciones  
 20 desde un sitio web, un servidor u otra fuente remota a través de un cable coaxial, una fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas como infrarrojos, radio y microondas, el cable coaxial, la fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas como infrarrojos, radio y microondas se incluyen en una definición del medio. Sin embargo, debe entenderse que el soporte de almacenamiento legible por ordenador y el soporte de almacenamiento de datos no incluyen conexiones,  
 25 portadoras, señales u otros medios transitorios, sino que en realidad significan soportes de almacenamiento tangibles no transitorios. Los discos y disquetes utilizados en esta memoria descriptiva incluyen un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD) y un disco Blu-ray. Los discos usualmente reproducen datos magnéticamente, mientras que los discos reproducen datos ópticamente con láseres. Las combinaciones de los anteriores elementos también deben incluirse en el alcance del medio legible  
 30 por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASICs), matrices de puerta programables de campo (FPGAs), u otros circuitos integrados equivalentes o  
 35 circuitos lógicos discretos. Por lo tanto, el término "procesador" utilizado en esta memoria descriptiva puede referirse a la estructura anterior, o cualquier otra estructura adecuada para implementar las tecnologías descritas en esta memoria descriptiva. Además, en algunos aspectos, las funciones descritas con referencia a los bloques lógicos ilustrativos, módulos y etapas descritos en esta memoria descriptiva pueden proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o pueden  
 40 incorporarse a un códec combinado. Además, todas las tecnologías pueden implementarse en uno o más circuitos o elementos lógicos.

Las tecnologías en esta solicitud pueden implementarse en diversos aparatos o dispositivos, incluyendo un auricular inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips).  
 45 Diversos componentes, módulos o unidades se describen en esta solicitud para enfatizar aspectos funcionales de los aparatos configurados para realizar las tecnologías descritas, pero no se implementan necesariamente mediante el uso de diferentes unidades de hardware. En realidad, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec en combinación con el software y/o firmware apropiado, o pueden ser proporcionadas por unidades de hardware interoperables (incluidos uno o  
 50 más procesadores descritos anteriormente).

En las realizaciones anteriores, la descripción de las realizaciones tienen enfoques respectivos. Para una parte que no se describe en detalle en una realización, consúltense las descripciones relacionadas en otras realizaciones.  
 55

Las descripciones anteriores son, simplemente, implementaciones de ejemplo específicas de esta solicitud, pero no pretenden limitar el alcance de protección de esta solicitud. Por lo tanto, el alcance de protección de esta solicitud estará sujeto al alcance de protección de las reivindicaciones.

**REIVINDICACIONES**

1. Un método de interpredicción, en donde un bloque a procesar comprende uno o más subbloques, y el método comprende:
- 5 determinar un vector de desplazamiento temporal, tempMV, del bloque a procesar basado en un bloque espacialmente vecino del bloque a procesar según lo siguiente:
- cuando availableFlagA<sub>1</sub> es igual a TRUE, se aplica lo siguiente:
- 10 - si son verdaderas todas las condiciones siguientes, tempMV es igual a mvL0A<sub>1</sub>:
- predFlagL0A<sub>1</sub> es igual a 1,
- 15 - DiffPicOrderCnt(ColPic, RefPicList[ 0 ][ reldxL0A<sub>1</sub> ]) es igual a 0;
- de lo contrario, si son verdaderas todas las condiciones siguientes, tempMV se establece igual a mvL1A<sub>1</sub>:
- slice\_type es igual a B,
- 20 - predFlagL1A<sub>1</sub> es igual a 1,
- DiffPicOrderCnt(ColPic, RefPicList[ 1 ][ reldxL1A<sub>1</sub> ]) es igual a 0;
- 25 en donde tempMV se utiliza para determinar un subbloque colocado del subbloque del bloque a procesar, en donde,
- availableFlagA<sub>1</sub> es el indicador de disponibilidad de una unidad codificadora vecina, A<sub>1</sub>, que es vecina a una unidad codificadora actual a la izquierda;
- 30 mvL0 A<sub>1</sub> es un vector de movimiento correspondiente a la lista 0 de A<sub>1</sub>;
- mvL1 A<sub>1</sub> es un vector de movimiento correspondiente a la lista 1 de A<sub>1</sub>;
- 35 predFlagL0A<sub>1</sub> es un indicador de dirección de predicción para la lista 0 de A<sub>1</sub>
- predFlagL1A<sub>1</sub> es un indicador de dirección de predicción para la lista 1 de A<sub>1</sub>;
- 40 DiffPicOrderCnt(ColPic, RefPicList[ 0 ][ reldxL0A<sub>1</sub> ]) es una función que determina una diferencia de recuento de orden de imagen entre una primera imagen, ColPic, que es la imagen colocada de la imagen actual, y una segunda imagen, RefPicList[ 0 ][ reldxL0A<sub>1</sub> ], que es una imagen de referencia de la lista 0, en donde, la RefPicList[ 0 ] es un conjunto de imágenes de referencia de la lista 0 y reldxL0A<sub>1</sub> es un índice de referencia de la lista 0 de A<sub>1</sub>;
- 45 DiffPicOrderCnt(ColPic, RefPicList[ 1 ][ reldxL1A<sub>1</sub> ]) es una función que determina una diferencia de recuento de orden de imagen entre la primera imagen, ColPic, y una tercera imagen, RefPicList[ 1 ][ reldxL1A<sub>1</sub> ], que es una imagen de referencia de la lista 1, en donde, la RefPicList[ 1 ] es un conjunto de imágenes de referencia de la lista 1 y reldxL1A<sub>1</sub> es un índice de referencia de la lista 1 de A<sub>1</sub>;
- 50 slice\_type es el tipo del segmento actual;
- en donde el método comprende además:
- 55 determinar un vector de movimiento del subbloque del bloque a procesar con base en un vector de movimiento del subbloque colocado, en donde el vector de movimiento del subbloque del bloque a procesar se obtiene en función de un primer vector de movimiento preestablecido cuando el vector de movimiento del subbloque colocado no está disponible.
- 60 2. Un aparato de interpredicción, en donde un bloque a procesar comprende uno o más subbloques, y el aparato comprende:
- un módulo de obtención de desplazamiento, configurado para determinar un vector de desplazamiento temporal, tempMV, del bloque a procesar basado en un bloque espacialmente vecino del bloque a procesar según lo siguiente:
- 65

cuando availableFlagA<sub>1</sub> es igual a TRUE, se aplica lo siguiente:

- si son verdaderas todas las condiciones siguientes, tempMV es igual a mvL0A<sub>1</sub>:

5 - predFlagL0A<sub>1</sub> es igual a 1,

- DiffPicOrderCnt(ColPic, RefPicList[ 0 ][ reldxL0A<sub>1</sub> ]) es igual a 0;

- de lo contrario, si son verdaderas todas las condiciones siguientes, tempMV se establece igual a mvL1A<sub>1</sub>:

10

- slice\_type es igual a B,

- predFlagL1A<sub>1</sub> es igual a 1,

15

- DiffPicOrderCnt(ColPic, RefPicList[ 1 ][ reldxL1A<sub>1</sub> ]) es igual a 0;

en donde tempMV se utiliza para determinar un subbloque colocado del subbloque del bloque a procesar,

en donde,

20

availableFlagA<sub>1</sub> es el indicador de disponibilidad de una unidad codificadora vecina, A<sub>1</sub>, que es vecina a una unidad codificadora actual a la izquierda;

mvL0 A<sub>1</sub> es un vector de movimiento correspondiente a la lista 0 de A<sub>1</sub>;

25

mvL1 A<sub>1</sub> es un vector de movimiento correspondiente a la lista 1 de A<sub>1</sub>;

predFlagL0A<sub>1</sub> es un indicador de dirección de predicción para la lista 0 de A<sub>1</sub>

30

predFlagL1A<sub>1</sub> es un indicador de dirección de predicción para la lista 1 de A<sub>1</sub>;

DiffPicOrderCnt(ColPic, RefPicList[ 0 ][ reldxL0A<sub>1</sub> ]) es una función que determina una diferencia de recuento de orden de imagen entre una primera imagen, ColPic, que es la imagen colocada de la imagen actual, y una segunda imagen, RefPicList[ 0 ][ reldxL0A<sub>1</sub> ], que es una imagen de referencia de la lista 0, en donde, la RefPicList[ 0 ] es un conjunto de imágenes de referencia de la lista 0 y reldxL0A<sub>1</sub> es un índice de referencia de la lista 0 de A<sub>1</sub>;

35

DiffPicOrderCnt(ColPic, RefPicList[ 1 ][ reldxL1A<sub>1</sub> ]) es una función que determina una diferencia de recuento de orden de imagen entre la primera imagen, ColPic, y una tercera imagen, RefPicList[ 1 ][ reldxL1A<sub>1</sub> ], que es una imagen de referencia de la lista 1, en donde, la RefPicList[ 1 ] es un conjunto de imágenes de referencia de la lista 1 y reldxL1A<sub>1</sub> es un índice de referencia de la lista 1 de A<sub>1</sub>;

40

slice\_type es el tipo del segmento actual;

45

en donde el aparato comprende además:

un módulo de obtención de vector de movimiento, configurado para determinar un vector de movimiento del subbloque del bloque a procesar con base en un vector de movimiento del subbloque colocado, en donde el vector de movimiento del subbloque del bloque a procesar se obtiene en función de un primer vector de movimiento preestablecido cuando el vector de movimiento del subbloque colocado no está disponible.

50

3. Un dispositivo de codificado de vídeo, que comprende una memoria no volátil y un procesador que se acoplan entre sí, en donde el procesador invoca el código de programa almacenado en la memoria, para realizar el método de interpretación según cualquiera de las reivindicaciones 1.

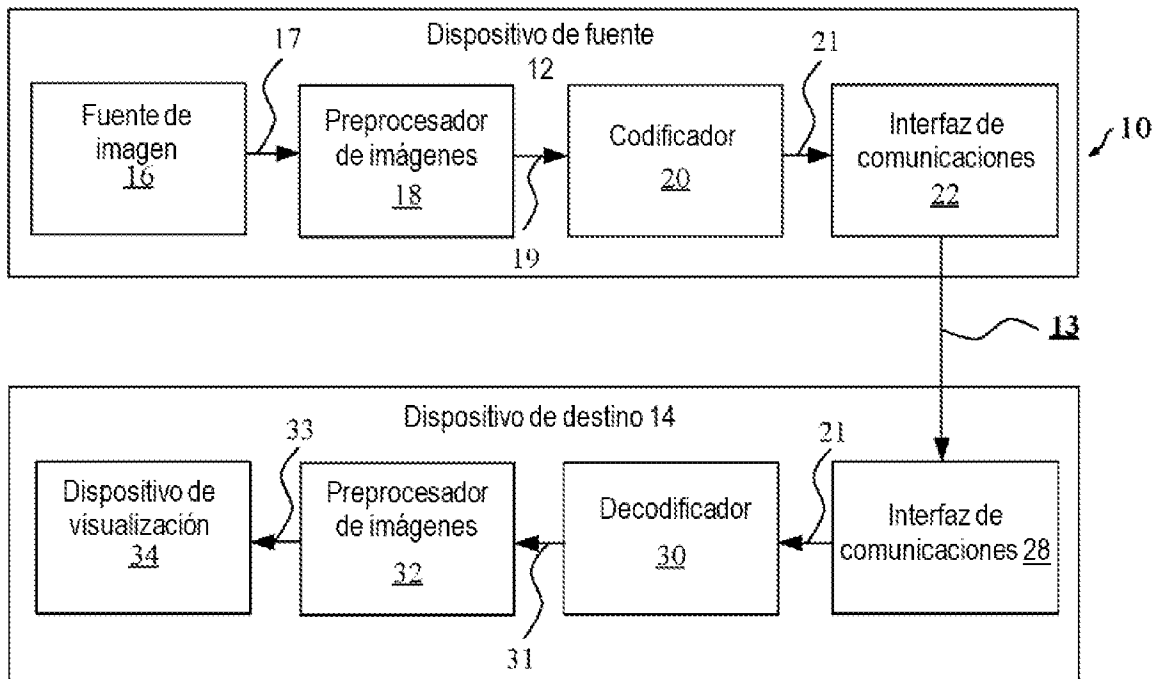


FIG. 1A

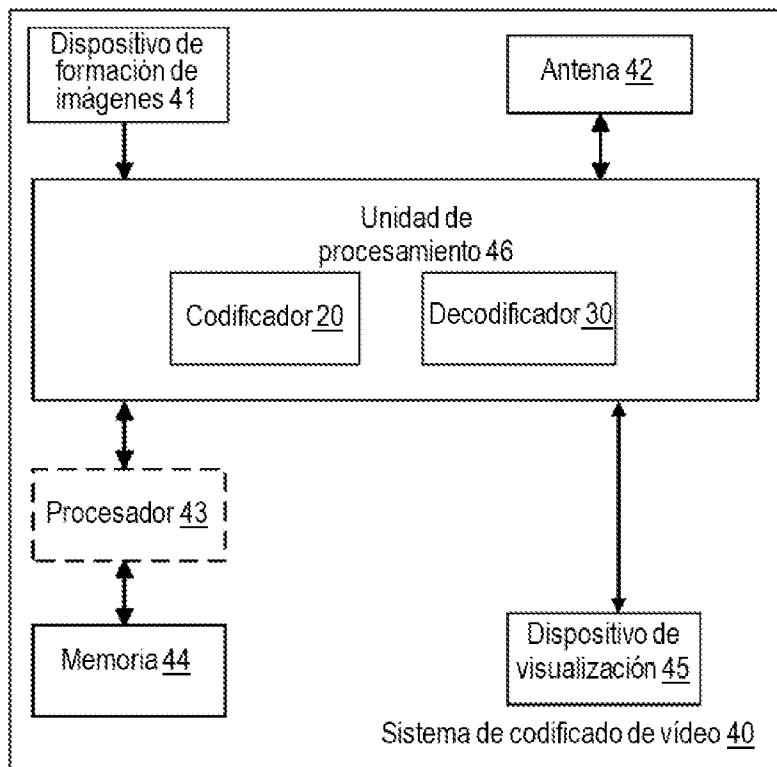


FIG. 1B



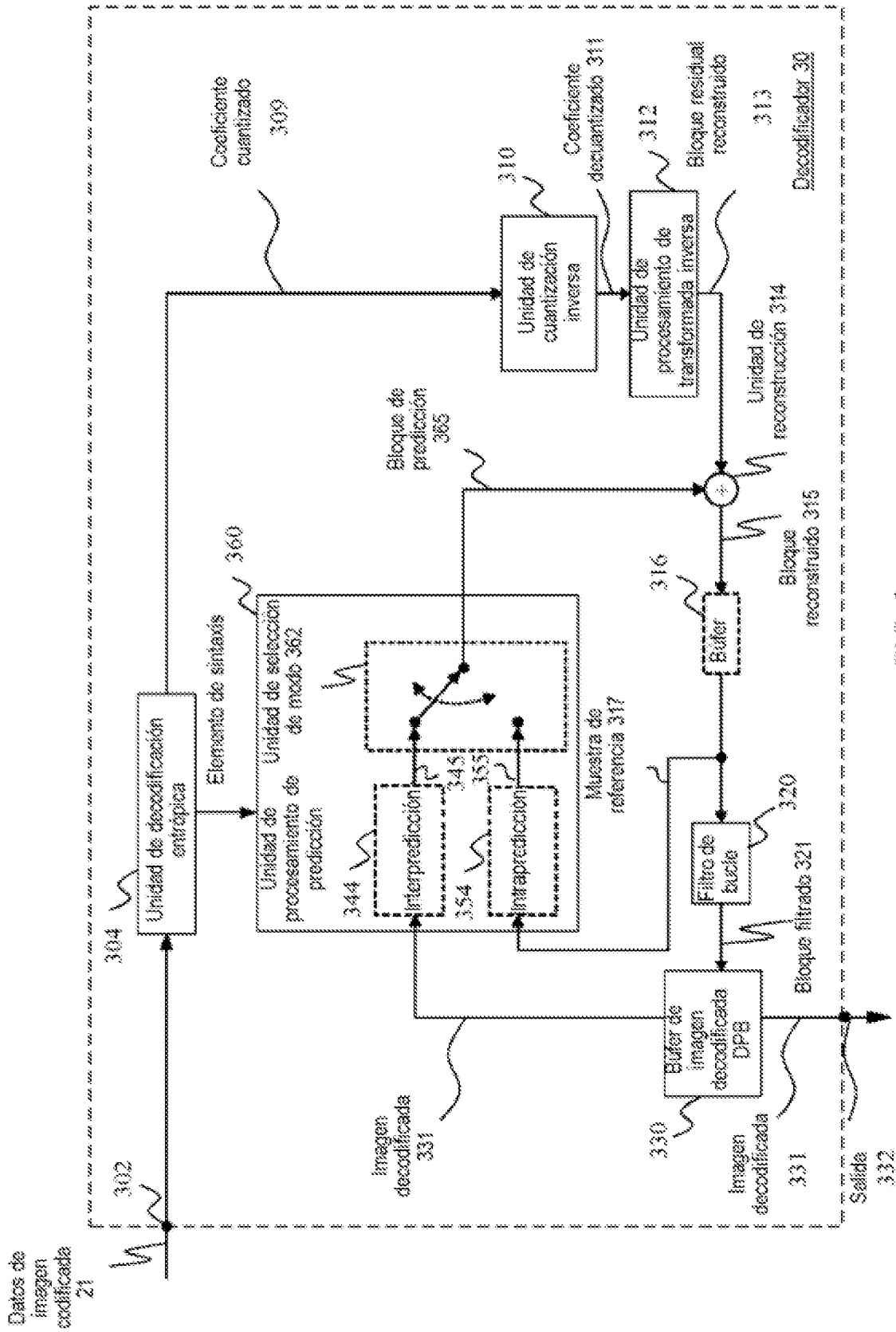


FIG. 3

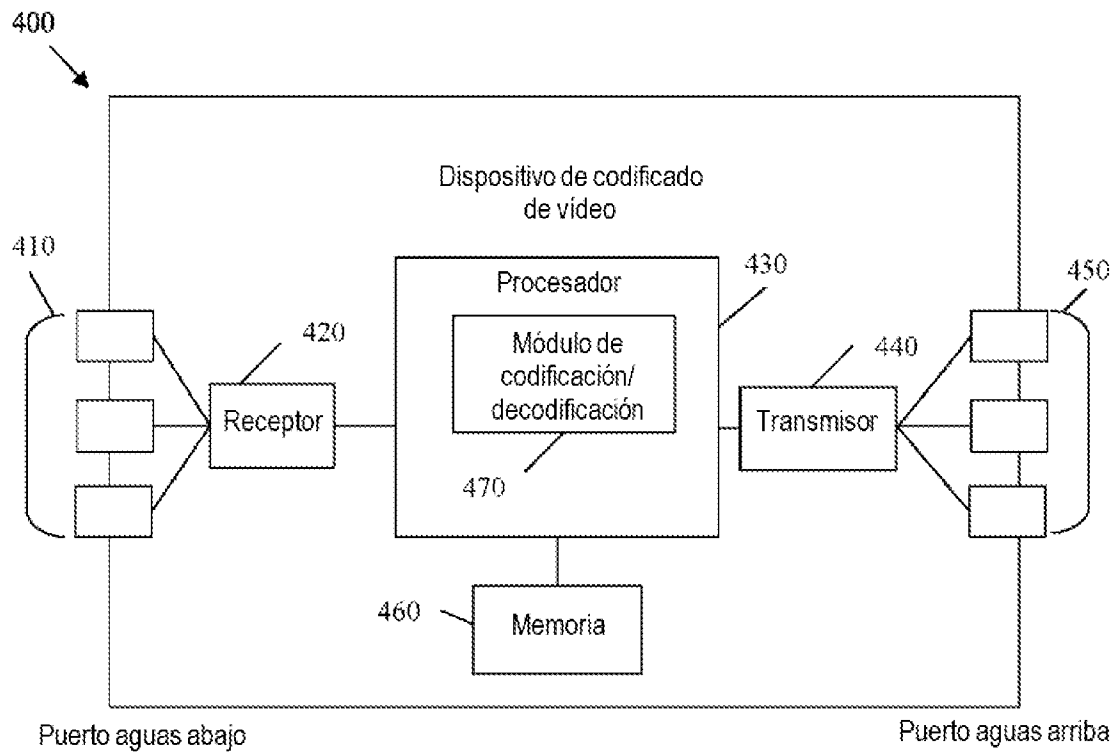


FIG. 4

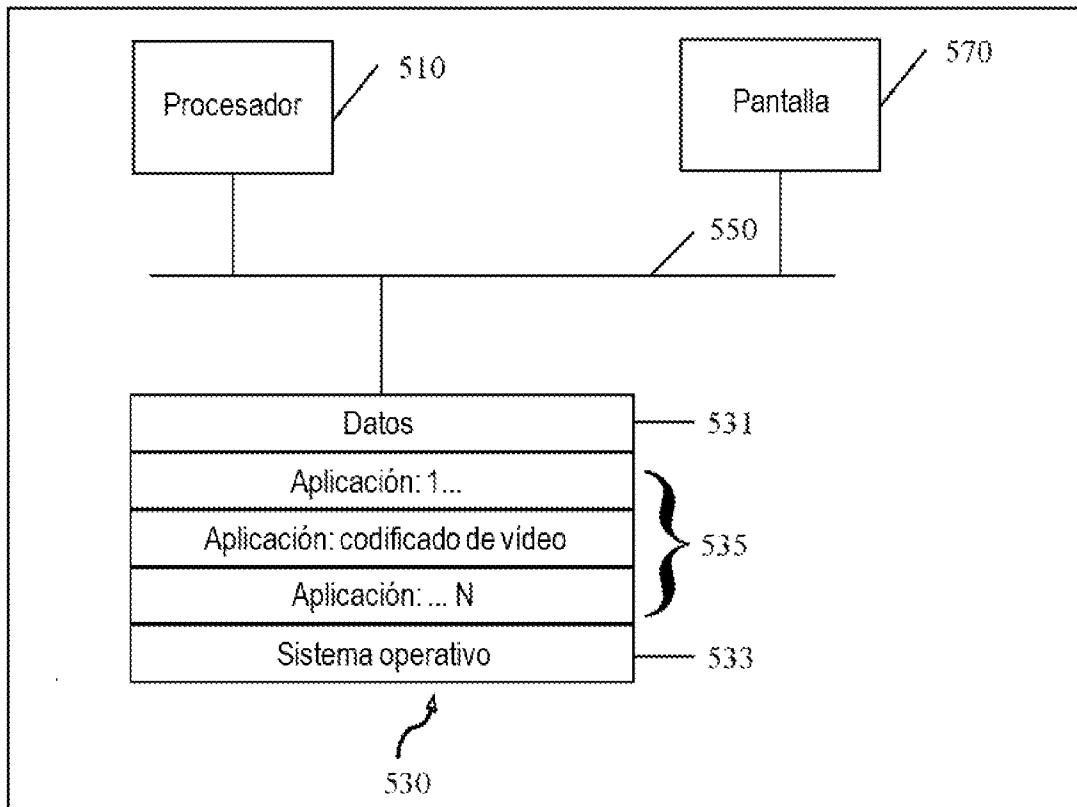


FIG. 5

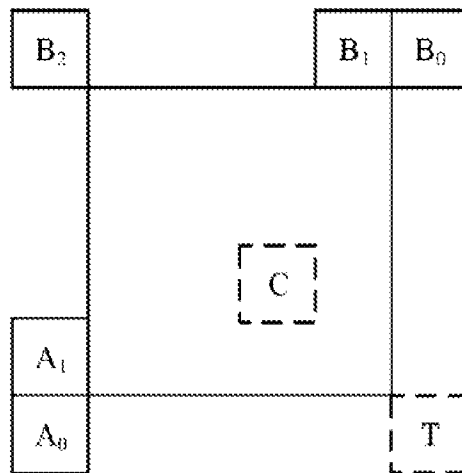


FIG. 6

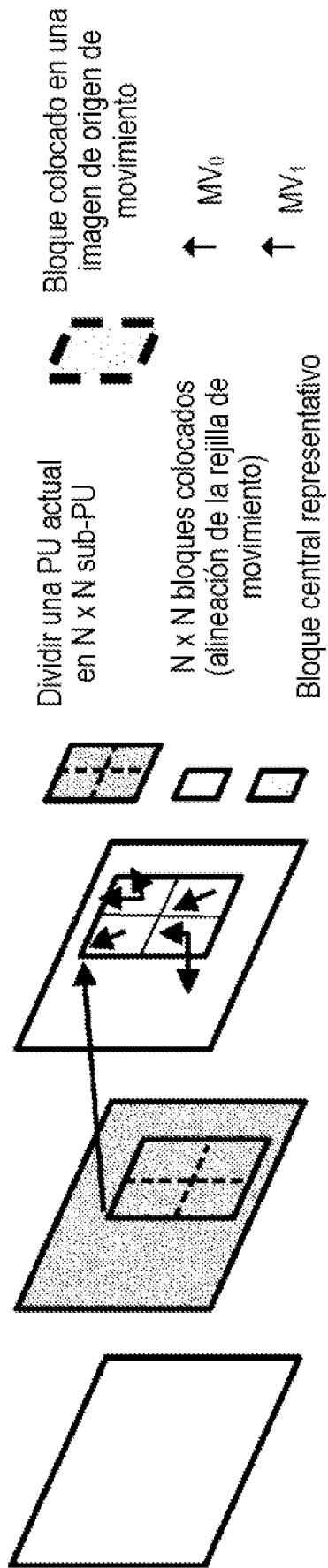


FIG. 7

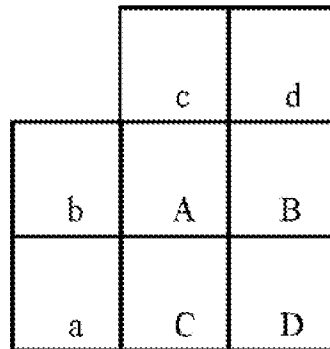


FIG. 8

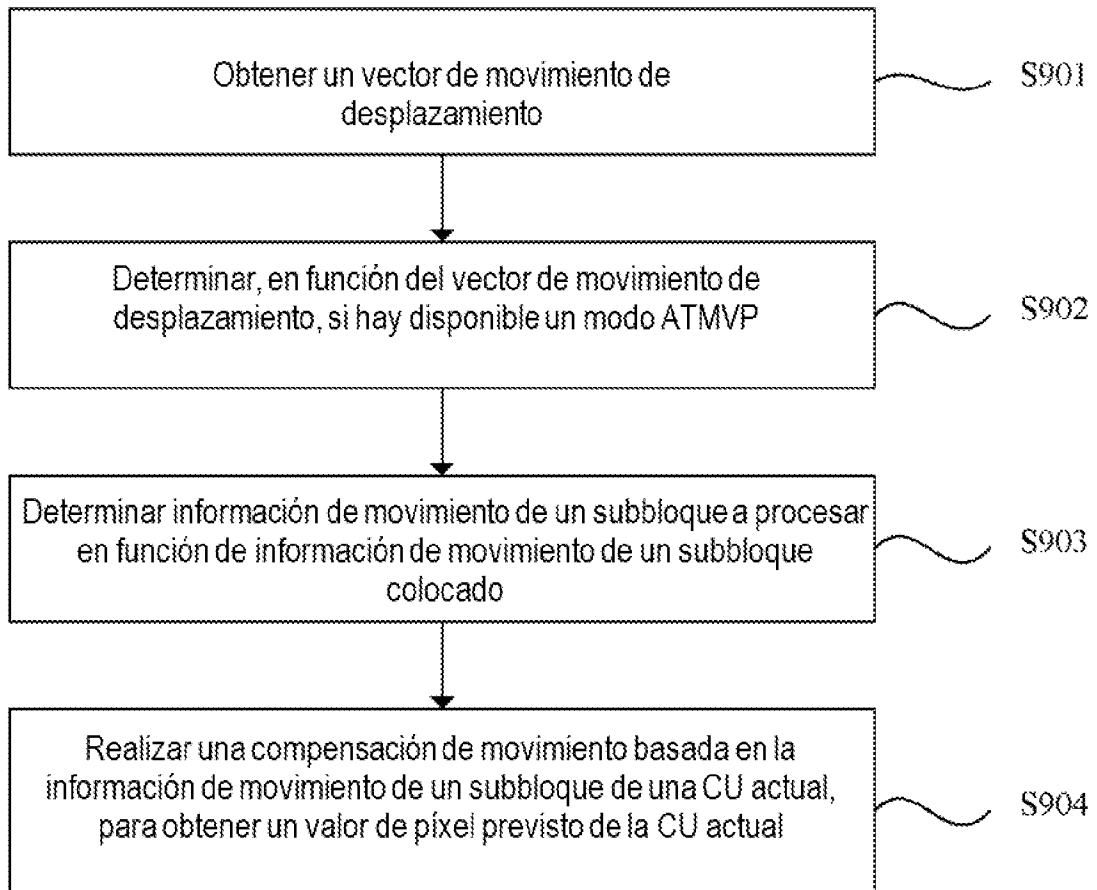


FIG. 9

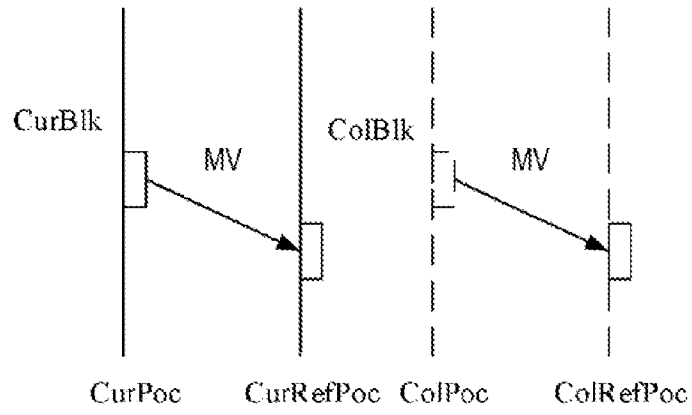


FIG. 10

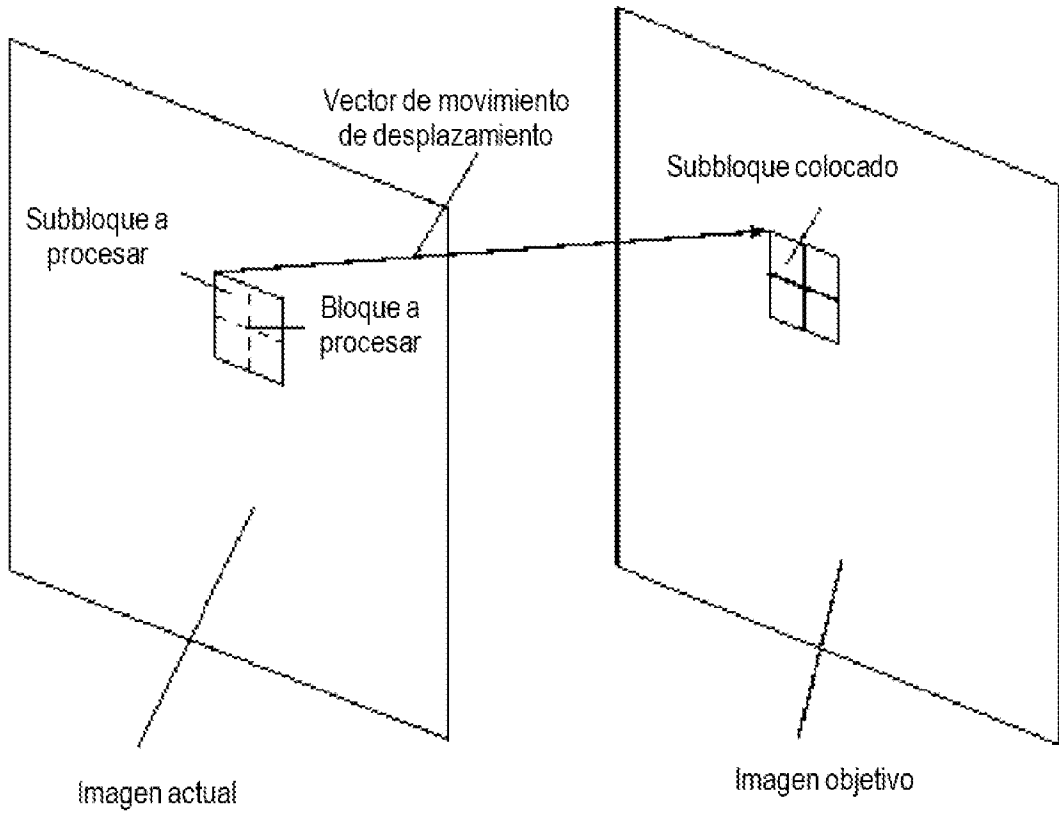


FIG. 11

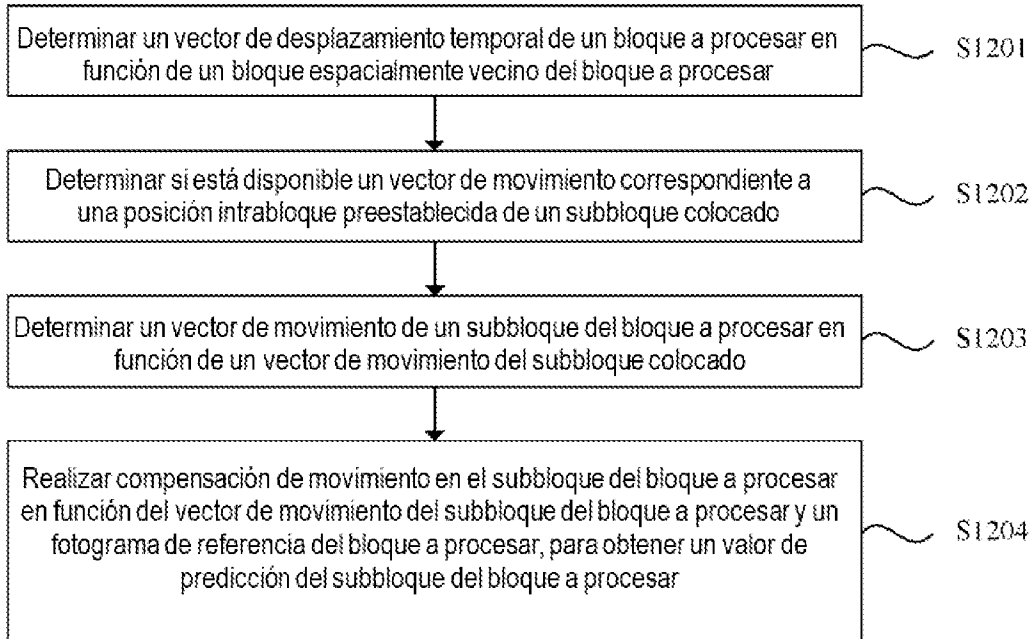


FIG. 12

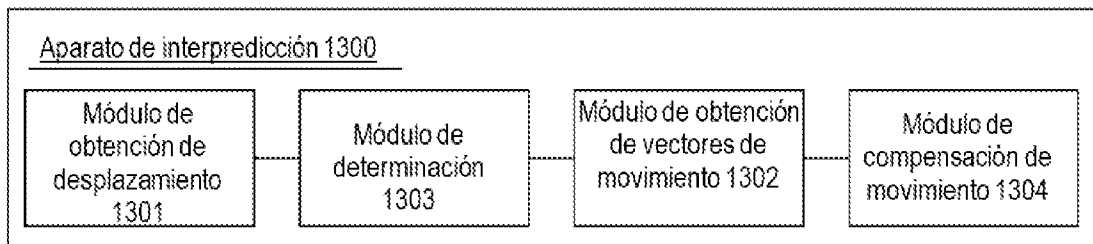


FIG. 13