

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 974 237**

51 Int. Cl.:

H04N 19/52 (2014.01)

H04N 19/70 (2014.01)

H04N 19/176 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **23.10.2019 PCT/CN2019/112749**

87 Fecha y número de publicación internacional: **07.05.2020 WO20088324**

96 Fecha de presentación y número de la solicitud europea: **23.10.2019 E 19879740 (9)**

97 Fecha y número de publicación de la concesión europea: **20.12.2023 EP 3852370**

54 Título: **Método y aparato de predicción de imágenes de vídeo**

30 Prioridad:

29.10.2018 CN 201811268188
29.12.2018 CN 201811642717

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
26.06.2024

73 Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building Bantian,
Longgang District
Shenzhen, Guangdong 518129, CN

72 Inventor/es:

CHEN, HUANBANG y
YANG, HAITAO

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 974 237 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 aparato de predicción de imágenes de vídeo

5 **CAMPO TÉCNICO**

Esta solicitud está relacionada con el campo de las tecnologías de codificado de imágenes, y, en particular, con un método y un aparato de predicción de imágenes de vídeo.

10 **ANTECEDENTES**

Con el desarrollo de tecnologías de información, se desarrollan rápidamente servicios de vídeo tales como televisión de alta definición, conferencias web, IPTV, y televisión 3D. Debido a ventajas tales como intuición y la alta eficiencia, las señales de vídeo se convierten en una manera principal de obtención de información en el día a día de las personas. Las señales de vídeo comprenden una gran cantidad de datos, y por lo tanto ocupan una gran cantidad de ancho de banda de transmisión y espacio de almacenamiento. Para transmitir y almacenar eficazmente las señales de vídeo, se tiene que realizar codificado de compresión necesita en las señales de vídeo. Una tecnología de compresión de vídeo se ha convertido gradualmente en una tecnología clave indispensable en el campo de aplicación de vídeo.

Un principio básico de la compresión por codificado de vídeo es reducir al máximo la redundancia usando correlaciones entre un dominio de espacio, un dominio de tiempo, y una palabra clave. Actualmente, un método prevalente es implementar compresión por codificado de vídeo usando un marco de trabajo de codificado de vídeo híbrido basado en bloques de imágenes y al realizar etapas tales como predicción (que incluye intrapredicción e interpredicción), transformada, cuantización y codificado por entropía.

En diversas soluciones de codificación/decodificación de vídeo, estimación de movimiento/compensación de movimiento en interpredicción es una tecnología clave que afecta el rendimiento de codificación/decodificación. En interpredicción existente, se añade predicción de vector de movimiento por fusión basada en subbloques en función de predicción por compensación de movimiento (MC) basada en bloques en la que se usa un modelo de movimiento traslacional. En la tecnología existente, no hay manera factible de determinar una longitud máxima de una lista de vectores de movimiento candidatos correspondiente a un modo de fusión de subbloque.

El último borrador de VVC se describe en el siguiente documento de la técnica anterior: BROSS B ET AL: "Versatile Video Coding (Draft 2)", 11. JVET MEETING; 20180711 - 20180718; LJUBLJANA; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG1 1 AND ITU-T SG.16), no. JVET-K1001 21 de septiembre de 2018 (2018-09-21), recuperado de internet: URL: http://phenix.int-evry.fr/jvet/doc_end_user/documents/11_Ljubljana/wg11/JVET-K1001-v6.zip JVET-K1001-v6.docx

40 **COMPENDIO**

Esta solicitud proporciona un método y un aparato de predicción de imágenes de vídeo, para proporcionar una manera de determinar una longitud máxima de una lista de vectores de movimiento candidatos en un modo de fusión de subbloque.

La presente invención se define en las reivindicaciones independientes. La divulgación habilitadora para la invención protegida se proporciona con las realizaciones descritas en relación con la figura 9.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

50 La FIGURA 1A es un diagrama de bloques de un ejemplo de un sistema de codificación y decodificación de vídeo 10 para implementar realizaciones de esta solicitud;
la FIGURA 1B es un diagrama de bloques de un ejemplo de un sistema de codificado de vídeo 40 para implementar realizaciones de esta solicitud;
55 la FIGURA 2 es un diagrama de bloques de una estructura de ejemplo de un codificador 20 para implementar realizaciones de esta solicitud;
la FIGURA 3 es un diagrama de bloques de una estructura de ejemplo de un decodificador 30 para implementar realizaciones de esta solicitud;
la FIGURA 4 es un diagrama de bloques de un ejemplo de un dispositivo de codificado de vídeo 400 para implementar realizaciones de esta solicitud;
60 la FIGURA 5 es un diagrama de bloques de un ejemplo de otro aparato de codificación o aparato de decodificación para implementar realizaciones de esta solicitud;
la FIGURA 6A es un diagrama esquemático de una ubicación candidata para información de movimiento para implementar realizaciones de esta solicitud;
65 la FIGURA 6B es un diagrama esquemático de una predicción de movimiento de punto de control heredado para implementar realizaciones de esta solicitud;

la FIGURA 6C es un diagrama esquemático de una predicción de movimiento de punto de control construido para implementar realizaciones de esta solicitud;

la FIGURA 6D es un diagrama esquemático de un procedimiento para combinar información de movimiento de punto de control para obtener información de movimiento de punto de control construido para implementar realizaciones de esta solicitud;

la FIGURA 6E es un diagrama esquemático de una manera de predicción de ATMVP para implementar realizaciones de esta solicitud;

la FIGURA 7 es un diagrama esquemático de una manera de predicción de vector de movimiento plano para implementar realizaciones de esta solicitud;

la FIGURA 8A es un diagrama de flujo de un método de interpredicción para implementar realizaciones de esta solicitud;

la FIGURA 8B es un diagrama esquemático de construir una lista de vector de movimiento candidato para implementar realizaciones de esta solicitud;

la FIGURA 8C es un diagrama esquemático de una unidad de compensación de movimiento para implementar realizaciones de esta solicitud;

la FIGURA 9 es un diagrama de flujo esquemático de un método de predicción de imágenes de vídeo para implementar realizaciones de esta solicitud;

la FIGURA 10 es un diagrama de flujo esquemático de otro método de predicción de imágenes de vídeo para implementar realizaciones de esta solicitud;

la FIGURA 11 es todavía un diagrama de flujo esquemático de todavía otro método de predicción de imágenes de vídeo para implementar realizaciones de esta solicitud;

la FIGURA 12A y la FIGURA 12B es un diagrama de flujo esquemático de incluso otro método de predicción de imágenes de vídeo para implementar realizaciones de esta solicitud;

la FIGURA 13 es un diagrama esquemático de un aparato 1300 para implementar realizaciones de esta solicitud;

la FIGURA 14 es un diagrama esquemático de un aparato 1400 para implementar realizaciones de esta solicitud; y

la FIGURA 15 es un diagrama esquemático de un aparato 1500 para implementar realizaciones de esta solicitud.

DESCRIPCIÓN DE REALIZACIONES

A continuación se describen las realizaciones de esta solicitud con referencia a los dibujos adjuntos en las realizaciones de esta solicitud. En la siguiente descripción, se hace referencia a los dibujos adjuntos que forman parte de esta divulgación y muestran, a modo de imagen, aspectos específicos de las realizaciones de esta solicitud o aspectos específicos en los que se pueden usar las realizaciones de esta solicitud. Debe entenderse que las realizaciones de esta solicitud se pueden usar en otros aspectos, y pueden comprender cambios estructurales o lógicos no representados en los dibujos adjuntos. Por lo tanto, las siguientes descripciones detalladas no se deben tomar en un sentido limitativo, y el alcance de esta solicitud se define por las reivindicaciones anexas. Por ejemplo, se debe entender una divulgación en relación con un método descrito también siguen siendo verdadera para un correspondiente dispositivo o sistema configurado para realizar el método y viceversa. Por ejemplo, si se describe una o más etapas de método específicas, un dispositivo correspondiente puede comprender una o más unidades tales como unidades funcionales, para realizar la una o más etapas de método descritas (por ejemplo, una unidad que realiza la una o más etapas; o una pluralidad de unidades cada una realizando una o más de la pluralidad de etapas), incluso si tal una o más unidades no se describen o ilustran explícitamente en los dibujos adjuntos. Por otro lado, por ejemplo, si se describe un aparato específico en función de una o más unidades tales como unidades funcionales, un método correspondiente puede comprender una etapa usada para realizar la funcionalidad de la una o más unidades (por ejemplo, una etapa usada para realizar la funcionalidad de la una o más unidades, o una pluralidad de etapas cada usado para realizar funcionalidad de una o más de una pluralidad de unidades), incluso si tal una o más etapas no se describen o ilustran explícitamente en los dibujos adjuntos. Además, se debe entender que las características de las diversas realizaciones y/u aspectos de ejemplo descritos en esta memoria descriptiva pueden combinarse entre sí, a menos que se indique específicamente lo contrario.

Las soluciones técnicas en las realizaciones de esta solicitud no son aplicables únicamente a un estándar de codificado de vídeo existente (por ejemplo, el estándar tal como H.264 o HEVC), sino también aplicable a un estándar futuro de codificado de vídeo (por ejemplo, el estándar H.266). Los términos usados en DESCRIPCIÓN DE REALIZACIONES de esta solicitud únicamente se usan para explicar realizaciones específicas de esta solicitud, pero no pretenden limitar esta solicitud. A continuación se describen primero brevemente conceptos relacionados en las realizaciones de esta solicitud.

El codificado de vídeo usualmente se refiere a procesamiento de una secuencia de imágenes que forman un vídeo o una secuencia de vídeo. El término "fotograma" o "imagen" se puede usar como sinónimos en el campo de codificado de vídeo. El codificado de vídeo en esta memoria descriptiva representa codificación de vídeo o decodificación de vídeo. El codificado de vídeo se realiza en un lado de fuente, y usualmente comprende procesar (por ejemplo, por compresión) una imagen de vídeo original para reducir una cantidad de datos requeridos para representar la imagen de vídeo, para almacenamiento y/o transmisión más eficientes. La decodificación de vídeo se realiza en un lado de

destino, y usualmente comprende procesamiento inverso comparado con un codificador para reconstruir una imagen de vídeo. El "codificado" de una imagen de vídeo en las realizaciones debe entenderse como "codificación" o "decodificación" de una secuencia de vídeo. Una combinación de una parte de codificación y una parte de decodificación también se denomina códec (codificación y decodificación).

5 Una secuencia de vídeo comprende una serie de imágenes (picture), una imagen se divide además en sectores (slice), y un sector se divide además en bloques (block). El procesamiento de codificado de vídeo se realiza por bloque. En algún nuevo estándar de codificado de vídeo, además se extiende un concepto de bloque. Por ejemplo, un macrobloque (MB) se introduce en el estándar H.264. El macrobloque puede además dividirse en una pluralidad de bloques de predicción (particiones) que se pueden usar para codificado predictivo. En el estándar de codificado de vídeo de alta eficiencia (HEVC), se usan conceptos básicos tales como unidad de codificado (CU), unidad de predicción (PU), y unidad de transformada (TU). Una pluralidad de unidades de bloque se obtienen a través de división funcional, y se describen usando una nueva estructura basada en árbol. Por ejemplo, una CU puede dividirse en CU más pequeñas en función de un árbol cuádruple, y la CU más pequeña puede dividirse aún más, para generar una estructura de árbol cuádruple. La CU es una unidad básica usado para dividir y codificar una imagen codificada. Una PU y una TU también tienen estructuras de árbol similares. La PU puede corresponder a un bloque de predicción, y es una unidad básica usado para codificado predictivo. La CU se divide además en una pluralidad de PU en función de un patrón de división. La TU puede corresponder a un bloque de transformada, y es una unidad básica usada para transformar una predicción residual. Sin embargo, en esencia, todas de la CU, la PU y la TU son conceptualmente bloques (o bloques de imágenes).

Por ejemplo, en HEVC, un CTV se divide en una pluralidad de CU usando una estructura de árbol cuádruple representada como árbol de codificado. Una decisión sobre si codificar un área de imagen usando predicción interimagen (temporal) o intraimagen (espacial) se hace a nivel de CU. Cada CU puede además dividirse en una, dos o cuatro PU en función de un patrón de división de PU. Dentro de un PU, se aplica un mismo proceso de predicción, e información relacionada se transmite a un decodificador en un base de PU. Tras obtener un bloque residual al aplicar el proceso de predicción basado en el patrón de división de PU, la CU puede particionarse en unidades de transformada (TU) en función de otra estructura de árbol cuádruple similar al árbol de codificado usado para la CU. En el reciente desarrollo de tecnologías de compresión de vídeo, un marco de partición de árbol cuádruple más árbol binario (Quadtree and binary tree, QTBT) se usa para particionar un bloque de codificado. En una estructura de bloque QTBT, la CU puede ser cuadrada o rectangular.

En esta memoria descriptiva, para facilitar la descripción y el entendimiento, un bloque de imagen a codificar en una imagen codificada actual puede denominarse bloque actual. Por ejemplo, en codificación, un bloque actual es un bloque actualmente siendo codificado; y en decodificación, un bloque actual es un bloque actualmente siendo decodificado. Un bloque de imagen decodificado, en una imagen de referencia, usado para predecir un bloque actual se denomina bloque de referencia. En otras palabras, un bloque de referencia es un bloque que proporciona una señal de referencia para un bloque actual. La señal de referencia representa un valor de píxel en el bloque de imagen. Un bloque que proporciona una señal de predicción para un bloque actual en una imagen de referencia 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, tras atravesar una pluralidad de bloques de referencia, se encuentra un bloque de referencia óptimo, y el bloque de referencia óptimo proporciona predicción para el bloque actual, y este bloque se denomina bloque de predicción.

45 En un caso de codificado de vídeo sin pérdidas, se pueden reconstruir imágenes de vídeo originales. Esto es, las imágenes de vídeo reconstruidas tienen la misma calidad que las imágenes de vídeo originales (asumiendo que no se provoca pérdida de transmisión u otra pérdida de datos durante el almacenamiento o la transmisión). En un caso de codificado de vídeo con pérdidas, se realiza compresión adicional a través de, por ejemplo, cuantización, para reducir una cantidad de datos requeridos para representar imágenes de vídeo, y las imágenes de vídeo no se pueden reconstruir completamente en un lado de decodificador. Esto es, la calidad de las imágenes de vídeo reconstruidas es menor o peor que la calidad de las imágenes de vídeo originales.

Varios estándares de codificado de vídeo H.261 son para "codecs de vídeo híbrido con pérdidas" (esto es, predicción espacial y temporal en un dominio de muestras se combina con codificado por transformada 2D para aplicar cuantización en un dominio de transformada). Cada imagen de una secuencia de vídeo usualmente se particiona en un conjunto de bloques no superpuestos, y usualmente se realiza codificado a nivel de bloque. En otras palabras, en un lado de codificador, un vídeo usualmente se procesa, esto es, codifica, a nivel de bloque (bloque de vídeo). Por ejemplo, un bloque de predicción se genera a través de predicción espacial (intraimagen) y predicción temporal (interimágenes), el bloque de predicción se sustrae de un bloque actual (un bloque actualmente siendo procesado o a procesar) para obtener un bloque residual, y el bloque residual se transforma y se cuantiza en el dominio de transformada para reducir una cantidad de datos que se van a transmitir (comprimir). En un lado de decodificador, procesamiento inverso comparado con el codificador se realiza en el bloque codificado o comprimido para reconstruir el bloque actual para representación. Es más, el codificador duplica un bucle de procesamiento de decodificador, de modo que el codificador y el decodificador generan las mismas predicciones (por ejemplo, intrapredicción y interpredicción) y/o reconstrucción para procesar, esto es, para codificar un subsiguiente bloque.

A continuación se describe una arquitectura de sistema usada en las realizaciones de esta solicitud. La FIGURA 1A es un diagrama de bloques esquemático de un ejemplo de un sistema de codificación y decodificación de vídeo 10 usado realizaciones de esta solicitud. Como se muestra en la FIGURA 1A, el sistema de codificación y decodificación de vídeo 10 puede comprender un dispositivo fuente 12 y un dispositivo de destino 14. El dispositivo fuente 12 genera datos de vídeo codificados, y por lo tanto el dispositivo 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 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 fuente 12, el dispositivo de destino 14, o ambos del dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender uno o más procesadores y una memoria acoplada al uno o más procesadores. La memoria puede comprender, pero sin limitación a esto, una RAM, una ROM, una EEPROM, una memoria flash, o cualquier otro medio que se pueda usar para almacenar código de programa deseado en forma de instrucción o una estructura de datos accesible por un ordenador, como se describe en esta memoria descriptiva. El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender diversos aparatos, incluido un ordenador de escritorio, un aparato informático móvil, un ordenador notebook (por ejemplo, un ordenador portátil), una tableta, un descodificador, un dispositivo manual telefónico tal como un denominado teléfono inteligente, una televisión, una cámara, un aparato de pantalla, un reproductor multimedia digital, una consola de videojuegos, un ordenador montado en vehículo, un dispositivo de comunicaciones inalámbricas, o algo semejante.

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

Una conexión de comunicación entre el dispositivo fuente 12 y el dispositivo de destino 14 puede implementarse en un enlace 13, y el dispositivo de destino 14 puede recibir datos de vídeo codificados del dispositivo fuente 12 en el enlace 13. El enlace 13 puede comprender uno o más medios o aparatos que pueden transferir los datos de vídeo codificados desde el dispositivo fuente 12 al dispositivo de destino 14. En un ejemplo, el enlace 13 puede comprender uno o más medios de comunicaciones que permiten al dispositivo fuente 12 transmitir directamente los datos de vídeo codificados al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo 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 pueden comprender un medio de comunicaciones inalámbricas y/o un medio de comunicaciones cableadas, por ejemplo, un espectro radiofrecuencia (RF) o uno o más cables de transmisión físicos. El uno o más medios de comunicaciones pueden ser 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 comunicaciones puede comprender un rúter, un conmutador (switch), una estación base, u otro dispositivo que facilita la comunicación desde el dispositivo fuente 12 al dispositivo de destino 14.

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

La fuente de imágenes 16 puede comprender o puede 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 un comentario (para codificación de contenido en pantalla, algún texto en una pantalla también se considera como parte de una imagen a codificar), por ejemplo, un procesador de gráficos de ordenador configurado para generar una imagen de animación por ordenador; o cualquier tipo de dispositivo configurado para obtener y/o proporcionar una imagen del mundo real o una imagen de animación por ordenador (por ejemplo, contenido en pantalla o una imagen de realidad virtual (VR)); y/o cualquier combinación de los mismos (por ejemplo, una imagen de realidad aumentada (AR)). La fuente de imágenes 16 puede ser una cámara configurada para capturar una imagen o una memoria configurada para almacenar una imagen. La fuente de imágenes 16 puede comprender además cualquier tipo de interfaz (interno o externo) a través de la que una imagen previamente capturada o generada se almacena y/o se obtiene o recibe una imagen. Cuando la fuente de imágenes 16 es una cámara, la fuente de imágenes 16 puede ser, por ejemplo, una cámara local, o una cámara integrado en el dispositivo fuente. Cuando la fuente de imágenes 16 es una memoria, la fuente de imágenes 16 puede ser una memoria local o, por ejemplo, una memoria integrada en el dispositivo fuente. Cuando la fuente de imágenes 16 comprende 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 imágenes externo tal como una cámara, una memoria externa, o un dispositivo externo generador de imágenes. El dispositivo externo generador de imágenes es, por ejemplo, un procesador externo de gráficos de ordenador, un ordenador, o un servidor. La interfaz puede ser cualquier tipo de interfaz, por ejemplo, una

interfaz cableada o inalámbrica o una interfaz óptica, según cualquier protocolo de interfaz en propiedad o estandarizado.

5 Una imagen puede considerarse como distribución bidimensional o matriz de elementos de píxel (elemento de imagen). El elemento de píxel en la distribución también se pueden denominar muestra. Una cantidad de muestras en direcciones (o ejes) horizontal y vertical de la distribución o la imagen define un tamaño y/o resolución de la imagen. Para representación de color, usualmente se emplean tres componentes de color, para ser específicos, la imagen puede representarse como o comprender tres distribuciones de muestra. Por ejemplo, en un formato RGB o un espacio de color, una imagen comprende distribuciones de muestras correspondientes de rojo, verde y azul. Sin embargo, en 10 codificado de vídeo, cada píxel usualmente se representa en un formato de luminancia/crominancia o un espacio de color. Por ejemplo, una imagen en un formato YUV comprende un componente de luminancia indicado por Y (a veces indicado por L en cambio) y dos componentes de crominancia indicados por U y V. El componente de luminancia (luma) Y representa brillo o intensidad de nivel de gris (por ejemplo, ambos son lo mismo en una imagen en escala de grises), y los dos componentes de crominancia (croma) U y V representan componentes de información de 15 crominancia o color. Correspondientemente, la imagen en el formato YUV comprende una distribución de muestras de luminancia de valores de muestra de luminancia (Y) y dos distribuciones de muestras de crominancia valores de crominancia (U y V). Imágenes en un formato RGB pueden transformarse o convertirse a un formato YUV y viceversa. Este proceso también se denomina conversión o transformación de color. Si una imagen es monocromática, la imagen puede comprender únicamente una distribución de muestras de luminancia. En esta realización de esta solicitud, una 20 imagen transmitida por la fuente de imágenes 16 al procesador de imágenes también se puede denominar datos de imagen sin procesar 17.

El preprocesador de imágenes 18 se configura para recibir los datos de imagen sin procesar 17 y realizar preprocesamiento en los datos de imagen sin procesar 17 para obtener una imagen preprocesada 19 o datos de 25 imagen preprocesados 19. Por ejemplo, el preprocesamiento realizado por el preprocesador de imágenes 18 puede comprender recortar, 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 denominado codificador de vídeo 20) se configura para recibir los datos de imagen preprocesados 19, y procesar los datos de imagen preprocesados 19 usando un modo de predicción relacionado (por ejemplo, un modo de predicción en cada realización de esta memoria descriptiva), para proporcionar datos de imagen codificados 21 (detalles estructurales del codificador 20 son como se describe adicionalmente más adelante en función de la FIGURA 2, la FIGURA 4 o la FIGURA 5). En algunas realizaciones, el codificador 20 se puede configurar para realizar 30 cada realización descrita más adelante, para implementar aplicación del lado de codificador de un método de predicción de bloque croma descrito en esta solicitud.

La interfaz de comunicaciones 22 se puede configurar para recibir los datos de imagen codificados 21, y transmitir los datos de imagen codificados 21 al dispositivo de destino 14 o cualquier otro dispositivo (por ejemplo, una memoria) por el enlace 13, para almacenamiento o reconstrucción directa. El otro dispositivo puede ser cualquier dispositivo 40 usado para decodificar o almacenamiento. La interfaz de comunicaciones 22 puede configurarse, por ejemplo, para empaquetar los datos de imagen codificados 21 en un formato apropiado, por ejemplo, un paquete de datos, para transmisión por el enlace 13.

El dispositivo de destino 14 comprende un decodificador 30. Opcionalmente, el dispositivo de destino 14 puede comprender además una interfaz de comunicaciones 28, un posprocesador de imágenes 32 y un dispositivo de 45 exposición 34. Se proporcionan descripciones por separado de la siguiente manera:

La interfaz de comunicaciones 28 se puede configurar para recibir los datos de imagen codificados 21 del dispositivo fuente 12 o cualquier otra fuente. La cualquiera otra fuente es, por ejemplo, un dispositivo de almacenamiento. El dispositivo de almacenamiento es, por ejemplo, un dispositivo de almacenamiento de datos de imagen codificados. La 50 interfaz de comunicaciones 28 se puede configurar para transmitir o recibir los datos de imagen codificados 21 por el enlace 13 entre el dispositivo fuente 12 y el dispositivo de destino 14 o en cualquier tipo de red. El enlace 13 es, por ejemplo, una conexión cableada directa o inalámbrica. El 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, por ejemplo, configurarse para desempaquetar el paquete 55 de datos transmitido a través de la interfaz de comunicaciones 22, para obtener los datos de imagen codificados 21.

Ambas de la interfaz de comunicaciones 28 y la interfaz de comunicaciones 22 se pueden configurar como interfaces de comunicaciones unidireccionales o interfaces de comunicaciones bidireccionales, y se pueden configurar para, por 60 ejemplo, enviar y recibir mensajes para configurar una conexión, y acusar recibo e intercambiar cualquier otra información relacionada con un enlace de comunicación y/o transmisión de datos tal como transmisión de datos de imagen codificados.

El decodificador 30 (o denominado decodificador 30) se configura para recibir los datos de imagen codificados 21 y proporcionar datos de imagen decodificados 31 o una imagen decodificada 31 (detalles estructurales del decodificador 30 son como se describe adicionalmente más adelante sobre la base de la FIGURA 3, FIGURA 4, o la FIGURA 5). En 65

algunas realizaciones, el decodificador 30 se puede configurar para realizar cada realización descrita más adelante, para implementar aplicación del lado de decodificador de un método de predicción de bloque croma descrito en esta solicitud.

5 El posprocesador de imágenes 32 se configura para posprocesar los datos de imagen decodificados 31 (también denominados datos de imagen reconstruida) para obtener datos de imagen posprocesados 33. El posprocesamiento realizado por el posprocesador de imágenes 32 puede comprender 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 imágenes 32 puede configurarse además para transmitir los datos de imagen posprocesados 33 al dispositivo de exposición 34.

10 El dispositivo de exposición 34 se configura para recibir los datos de imagen posprocesados 33 para exponer una imagen, por ejemplo, a un usuario o un espectador. El dispositivo de exposición 34 puede ser o puede comprender cualquier tipo de pantalla para presentar una imagen reconstruida, por ejemplo, una pantalla o monitor integrados o externos. Por ejemplo, la pantalla puede comprender una pantalla de cristal líquido (LCD), una pantalla de diodos emisores de luz orgánicos (OLED), una pantalla de plasma, un proyector, una pantalla de microleds, un cristal líquido en silicio (LCoS), un procesador de luz digital (DLP), o cualquier otro tipo de pantalla.

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

20 Como será evidente para un experto en la técnica sobre la base de las descripciones, la existencia y la división (exacta) de funcionalidades de diferentes unidades o funcionalidades del dispositivo fuente 12 y/o el dispositivo de destino 14 mostrados en la FIGURA 1A pueden variar dependiendo de un dispositivo de real y la aplicación. El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de un gran abanico de dispositivos, incluido cualquier tipo de dispositivo de mano o estacionario, por ejemplo, un ordenador notebook u ordenador portátil, un teléfono móvil, un teléfono inteligente, una tableta o tablet, una videocámara, un ordenador de escritorio, un decodificador, un televisor, una cámara, un dispositivo montado en vehículo, un dispositivo de exposición, un reproductor multimedia digital, una consola de videojuegos, un dispositivo de retrasmisión de vídeo (tal como un servidor de servicios de contenidos o un servidor de entrega de contenidos), un dispositivo receptor de difusión, o un dispositivo transmisor de difusión, y puede usar o no usar cualquier tipo de sistema operativo.

30 El codificador 20 y el decodificador 30 pueden implementarse cada uno como cualquiera de diversos circuitos apropiados, por ejemplo, uno o más microprocesadores, procesadores de señales digitales (digital signal processor, DSP), circuitos integrados de aplicación específica (Application-Specific Integrated Circuit, ASIC), matrices de puertas programables en campo (Field-Programmable Gate Array, FPGA), lógica discreta, hardware, o cualquier combinación de los mismos. Si las tecnologías se implementan parcialmente usando software, un dispositivo puede almacenar una instrucción de software en un soporte de almacenamiento legible por ordenador no transitorio adecuado y puede ejecutar la instrucción usando hardware tal como uno o más procesadores, para realizar las tecnologías de esta divulgación. Cualquier contenido anterior (incluido hardware, software, una combinación de hardware y software, y similares) puede considerarse uno o más procesadores.

40 En algunos casos, el sistema de codificación y decodificación de vídeo 10 mostrado en la FIGURA 1A es meramente un ejemplo, y las tecnologías de esta solicitud son aplicables a entornos de codificado de vídeo (por ejemplo, codificación de vídeo o decodificación de vídeo) que no necesariamente comprenden comunicación de datos entre un dispositivo de codificación y un dispositivo de decodificación. En otro ejemplo, pueden recuperarse datos de una memoria local, transmitirse por una red, o algo semejante. Un dispositivo de codificación de vídeo puede codificar datos y almacenar los datos en una memoria y/o un dispositivo de decodificación de vídeo puede recuperar datos de la memoria y decodificar los datos. En algunos ejemplos, codificación y decodificación se realizan por dispositivos que no se comunican entre sí, sino que simplemente codifican datos en una memoria y/o recuperan datos de la memoria y decodifican los datos.

50 La FIGURA 1B es un diagrama que ilustra un ejemplo de un sistema de codificado de vídeo 40 que incluye el codificador 20 en la FIGURA 2 y/o el decodificador 30 en la FIGURA 3 según una realización de ejemplo. El sistema de codificado de vídeo 40 puede implementar una combinación de diversas tecnologías en las realizaciones de esta solicitud. En una implementación ilustrada, el sistema de codificado de vídeo 40 puede comprender un dispositivo de formación de imágenes 41, el codificador 20, el decodificador 30 (y/o un codificador/decodificador de vídeo implementado por 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 exposición 45.

65

Como se muestra en la FIGURA 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 exposición 45 pueden comunicarse entre sí. Como se describe, aunque el sistema de codificado de vídeo 40 se ilustra con ambos del codificador 20 y el decodificador 30, en diferentes ejemplos, el sistema de codificado de vídeo 40 puede comprender únicamente el codificador 20 o únicamente el decodificador 30.

En algunos ejemplos, la antena 42 se puede configurar para transmitir o recibir un flujo de bits de datos de vídeo codificado. Además, en algunos ejemplos, el dispositivo de exposición 45 se puede configurar para presentar los datos de vídeo. En algunos ejemplos, el circuito lógico 47 puede implementarse por la unidad de procesamiento 46. La unidad de procesamiento 46 puede comprender una lógica de circuito integrado de aplicación específica (ASIC), un procesador de gráficos, un procesador de finalidad general, o algo semejante. El sistema de codificado de vídeo 40 también puede comprender el procesador opcional 43. De manera semejante, el procesador opcional 43 puede comprender lógica de circuito integrado de aplicación específica (ASIC), un procesador de gráficos, un procesador de finalidad general, o algo semejante. En algunos ejemplos, el circuito lógico 47 puede ser implementado por hardware, por ejemplo, hardware dedicado de codificado de vídeo, y el procesador 43 puede implementarse usando software de finalidad general, un sistema operativo, o algo semejante. Adicionalmente, la memoria 44 puede ser cualquier tipo de memoria, por ejemplo, una memoria volátil (por ejemplo, una memoria de acceso aleatorio estática (SRAM) o una memoria de acceso aleatorio dinámica (DRAM)), o una memoria no volátil (por ejemplo, una memoria flash). En un ejemplo no limitativo, la memoria 44 puede ser implementada por una memoria caché. En algunos ejemplos, el circuito lógico 47 puede acceder a la memoria 44 (por ejemplo, para implementación de un búfer de imágenes). En otros ejemplos, el circuito lógico 47 y/o la unidad de procesamiento 46 pueden comprender una memoria (por ejemplo, una caché) para implementación de un búfer de imágenes o algo semejante.

En algunos ejemplos, el codificador 20 implementado por el circuito lógico puede comprender un búfer de imágenes (por ejemplo, implementado por la unidad de procesamiento 46 o la memoria 44) y una unidad de procesamiento de gráficos (por ejemplo, implementada por 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 comprender el codificador 20 implementado por el circuito lógico 47, para implementar diversos módulos que se describen con referencia a la FIGURA 2 y/o cualquier otro sistema o subsistema codificador descritos en esta memoria descriptiva. El circuito lógico se puede configurar para realizar diversas operaciones descritas en esta memoria descriptiva.

En algunos ejemplos, el decodificador 30 puede ser implementado por el circuito lógico 47 de manera similar, para implementar diversos módulos que se describen con referencia a un decodificador 30 en la FIGURA 3 y/o cualquier otro sistema o subsistema decodificadores descritos en esta memoria descriptiva. En algunos ejemplos, el decodificador 30 implementado por el circuito lógico puede comprender un búfer de imágenes (por ejemplo, implementado por la unidad de procesamiento 2820 o la memoria 44) y una unidad de procesamiento de gráficos (por ejemplo, implementada por 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 comprender el decodificador 30 implementado por el circuito lógico 47, para implementar diversos módulos que se describen con referencia a la FIGURA 3 y/o cualquier otro sistema o subsistema de decodificador descritos en esta memoria descriptiva.

En algunos ejemplos, la antena 42 se puede configurar para recibir un flujo de bits de datos de vídeo codificado. Como se describe, el flujo de bits codificado puede comprender datos, un indicador, un valor de índice, datos de selección de modo y similares relacionados con codificación de fotogramas de vídeo descrita en esta memoria descriptiva, por ejemplo, datos relacionados con particionamiento 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 el particionamiento de codificado). El sistema de codificado de vídeo 40 puede comprender además el decodificador 30 acoplado a la antena 42 y configurado para decodificar el flujo de bits codificado. El dispositivo de exposición 45 se configura para presentar un fotograma de vídeo.

Debe entenderse que, en esta realización de esta solicitud, para el ejemplo descrito con referencia al codificador 20, el decodificador 30 se puede configurar para realizar un proceso inverso. Con relación a señalar elemento de sintaxis, el decodificador 30 se puede configurar para recibir y analizar sintácticamente este tipo de elemento de sintaxis y correspondientemente decodificar datos de vídeo relacionados. En algunos ejemplos, el codificador 20 puede codificar por entropía el elemento de sintaxis en un flujo de bits de vídeo codificado. En tales ejemplos, el decodificador 30 puede analizar sintácticamente el elemento de sintaxis y correspondientemente decodificar datos de vídeo relacionados.

Se debe observar que un método descrito en las realizaciones de esta solicitud se usa principalmente en un proceso de interpredicción. Este proceso es realizado por ambos del codificador 20 y el decodificador 30. El codificador 20 y el decodificador 30 en las realizaciones de esta solicitud pueden ser, por ejemplo, un codificador y un decodificador correspondientes a un protocolo de estándar de vídeo tal como H.263, H.264, HEVC, MPEG-2, MPEG-4, VP8, o VP9, o un protocolo de estándar de vídeo de próxima generación (tal como H.266).

La FIGURA 2 es un diagrama de bloques esquemático/conceptual de un ejemplo de un codificador 20 para implementar realizaciones de esta solicitud. En el ejemplo de la FIGURA 2, el codificador 20 comprende 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 de entropía 270. La unidad de procesamiento de predicción 260 puede comprender una unidad de interpretación 244, una unidad de intrapredicción 254 y una unidad de selección de modo 262. La unidad de interpretación 244 puede comprender una unidad de estimación de movimiento y una unidad de compensación de movimiento (no se muestran). El codificador 20 mostrado en la FIGURA 2 también se pueden denominar codificador de vídeo híbrido o un codificador de vídeo según un códec de vídeo híbrido.

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 de entropía 270 forman una ruta de señal directa del codificador 20, mientras 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, y 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 (haciendo referencia a un decodificador 30 en la FIGURA 3).

El codificador 20 recibe, por ejemplo, por medio de un entrada 202, una imagen 201 o un bloque de imagen 203 de una 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 se puede denominar bloque de imagen actual o bloque de imagen a codificar. 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 de una misma secuencia de vídeo, esto es, la secuencia de vídeo que también comprende la imagen actual).

Una realización del codificador 20 puede comprender una unidad de particionamiento (que no se representa en la FIGURA 2), configurada para particionar la imagen 201 en una pluralidad de bloques tales como bloques de imágenes 203. La imagen 201 usualmente se particiona en una pluralidad de bloques no superpuestos. La unidad de particionamiento se puede configurar para usar un mismo tamaño de bloque para todas las imágenes en una secuencia de vídeo y una correspondiente cuadrícula que define el tamaño de bloque, o cambiar un tamaño de bloque entre imágenes o subconjuntos o grupos de imágenes, y particionar cada imagen en correspondientes bloques.

En un ejemplo, la unidad de procesamiento de predicción 260 en el codificador 20 se puede configurar para realizar cualquier combinación de las tecnologías de particionamiento descritas anteriormente.

Como la imagen 201, el bloque de imagen 203 también es o puede ser considerado como distribución o matriz bidimensional de muestras con valores de muestra, aunque un tamaño del bloque de imagen 203 es menor que un tamaño de la imagen 201. En otras palabras, el bloque de imagen 203 puede comprender, por ejemplo, una distribución de muestras (por ejemplo, una distribución luma en un caso de una imagen monocromática 201), tres distribuciones de muestras (por ejemplo, una distribución luma y dos distribuciones croma en un caso de una imagen a color), o cualquier otra cantidad y/o tipo de distribuciones dependiendo de un formato de color aplicado. Una cantidad de muestras en direcciones (o ejes) horizontal y vertical del bloque de imagen 203 define un tamaño del bloque de imagen 203.

El codificador 20 mostrado en la FIGURA 2 se configura para codificar la imagen 201 bloque por bloque. Por ejemplo, codificación y predicción se realizan por bloque de imagen 203.

La unidad de cálculo residual 204 se configura para calcular un bloque residual 205 en función del bloque de imagen 203 y un bloque de predicción 265 (además detalles acerca del bloque de predicción 265 se proporcionan a continuación), por ejemplo, al restar valores de muestra del bloque de predicción 265 de valores de muestra del bloque de imagen 203 muestra por muestra (píxel por píxel), para obtener el bloque residual 205 en un dominio de muestras.

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), para valores de muestra del bloque residual 205 para obtener coeficientes de transformada 207 en un dominio de transformadas. El coeficiente de transformada 207 también se pueden denominar coeficiente residual de transformada y representa el bloque residual 205 en el dominio de transformada.

La unidad de procesamiento de transformada 206 se puede configurar para aplicar aproximaciones enteras de DCT/DST, tales como transformadas especificadas en HEVC/H.265. En comparación con una transformada DCT ortogonal, tales aproximaciones enteras usualmente se escalan en función de un factor. Para preservar una norma de un bloque residual que se procesa usando transformadas hacia delante e inversas, se aplica un factor de escala adicional como parte del proceso de transformada. El factor de escala se selecciona usualmente en función de algunas restricciones, por ejemplo, el factor de escala que es una potencia de dos para una operación de desplazamiento, una profundidad de bits del coeficiente de transformada, y una compensación entre precisión y costes de implementación. Por ejemplo, un factor de escala específico se especifica para la transformada inversa por, por ejemplo, la unidad de

procesamiento de transformada inversa 212 en el lado del decodificador 30 (y una correspondiente transformada inversa por, por ejemplo, la unidad de procesamiento de transformada inversa 212 en el lado del codificador 20), y correspondientemente, un factor de escala correspondiente puede especificarse para la transformada hacia delante por la unidad de procesamiento de transformada 206 en el lado del codificador 20.

5 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 la profundidad de bits relacionada con algunos 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, donde 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 usar escalas diferentes para lograr una cuantización más fina o más basta. El tamaño de etapa de cuantización más pequeño corresponde a una cuantización más fina, y un tamaño de etapa de cuantización más grande corresponde a una cuantización más basta. Un tamaño de etapa de cuantización apropiado puede indicarse mediante un parámetro de cuantización (QP). Por ejemplo, el parámetro de cuantización puede ser un índice de un conjunto predefinido de tamaños de etapa de cuantización apropiados. Por ejemplo, un parámetro de cuantización más pequeño puede corresponder a una cuantización más fina (tamaño de etapa de cuantización más pequeño) y un parámetro de cuantización más grande puede corresponder a una cuantización más basta (tamaño de etapa de cuantización más grande) o viceversa. La cuantización puede comprender división por un tamaño de etapa de cuantización y cuantización correspondiente o cuantización inversa, por ejemplo, realizada por la unidad de cuantización inversa 210, o puede comprender multiplicación por un tamaño de etapa de cuantización. En realizaciones según algunos estándares tales como HEVC, puede usarse un parámetro de cuantización para determinar el tamaño de etapa de cuantización. Generalmente, el tamaño de la etapa de cuantización se puede calcular sobre la base de un parámetro de cuantización usando una aproximación de punto fijo de una ecuación que incluye división. Se pueden introducir factores de escala adicionales para la cuantización y descuantización para restaurar la norma del bloque residual, donde la norma del bloque residual puede modificarse debido a la escala utilizada en la aproximación de punto fijo de la ecuación para el tamaño de etapa de cuantización y el parámetro de cuantización. En una implementación de ejemplo, una escala de la transformada inversa puede combinarse con una escala de la descuantización. Alternativamente, puede usarse una tabla de cuantización personalizada y señalizarse desde un codificador a un decodificador, por ejemplo en un flujo de bits. La cuantización es una operación con pérdidas, donde la pérdida aumenta con el aumento del tamaño de etapa de cuantización.

35 La unidad de cuantización inversa 210 se configura para aplicar la cuantización inversa de la unidad de cuantización 208 al coeficiente cuantizado para obtener un coeficiente descuantizado 211, por ejemplo, aplicar, sobre la base o usando un mismo tamaño de 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 descuantizado 211 también pueden denominarse coeficiente residual descuantizado 211 y corresponden al coeficiente de transformada 207, aunque el coeficiente descuantizado 211 usualmente es diferente del coeficiente de transformada debido a una pérdida provocada por la cuantización.

45 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 (discrete sine transform, DST), para obtener un bloque de transformada inversa 213 en el dominio de muestra. El bloque de transformada inversa 213 también se puede denominar bloque descuantizado de transformada inversa 213 o bloque residual de transformada inversa 213.

50 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, por ejemplo, añadiendo valores de muestra del bloque residual reconstruido 213 y los valores de muestra del bloque de predicción 265, para obtener un bloque reconstruido 215 en el dominio de muestras.

55 Opcionalmente, una unidad de búfer 216 ("búfer" 216 por abreviar) de, por ejemplo, el búfer de línea 216, se configura para almacenar temporalmente o almacenar el bloque reconstruido 215 y un correspondiente valor de muestra, por ejemplo, para intrapredicción. En otras realizaciones, el codificador se puede configurar para usar un bloque reconstruido no filtrado y/o un correspondiente valor de muestra que se almacenan en la unidad de búfer 216 para realizar cualquier tipo de estimación y/o predicción, por ejemplo, intrapredicción.

60 Por ejemplo, en una realización, el codificador 20 se puede configurar de modo que la unidad de búfer 216 se configura para almacenar el bloque reconstruido 215 no únicamente usado para la unidad de intrapredicción 254 sino también usado para la unidad de filtro de bucle 220 (que no se muestra en la FIGURA 2) y/o de modo que, por ejemplo, la unidad de búfer 216 y la unidad de búfer de imágenes decodificadas 230 forman un búfer. En otras realizaciones, un bloque filtrado 221 y/o un bloque o muestra (que no se muestra en la FIGURA 2) del búfer de imágenes decodificadas 230 se usa como entrada o base para la unidad de intrapredicción 254.

65

La unidad de filtro de bucle 220 ("filtro de bucle" 220 por abreviar) se configura para filtrar el bloque reconstruido 215 para obtener un bloque filtrado 221, para suavizar transiciones de píxeles o mejorar la calidad de vídeo. La unidad de filtro de bucle 220 está pensada para representar uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de desplazamiento adaptativo por muestra (SAO), u otros filtros, por ejemplo, un filtro bilateral, un filtro de bucle adaptativo (ALF), un filtro de agudizamiento o suavización, o un filtro colaborativo. Aunque la unidad de filtro de bucle 220 se muestra como filtro en bucle en la FIGURA 2, en otra implementación, la unidad de filtro de bucle 220 puede implementarse como filtro posbucle. El bloque filtrado 221 también se puede denominar 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 realiza una operación de filtrado en el bloque codificado reconstruido.

En una realización, el codificador 20 (correspondientemente, la unidad de filtro de bucle 220) se puede configurar para tener como salida un parámetro de filtro de bucle (por ejemplo, información de desplazamiento adaptativo de muestras), por ejemplo, directamente o tras codificación por entropía realizada por la unidad de codificación por entropía 270 o cualquier otra unidad de codificación por entropía, de modo que el decodificador 30 puede 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 imagen de referencia para usar en datos de codificación de vídeo por el codificador 20. El DPB 230 puede formarse por cualquiera de una variedad de dispositivos de almacenamiento, tales como memoria de acceso aleatorio dinámica (DRAM), incluida DRAM sincrónica (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de almacenamiento. La DPB 230 y el búfer 216 pueden ser proporcionados por un mismo dispositivo de almacenamiento o dispositivos de almacenamientos 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 otros bloques filtrados previamente, por ejemplo, bloques reconstruidos y filtrados previamente 221, de la misma imagen actual o de imágenes diferentes, por ejemplo, imágenes previamente reconstruidas, y puede proporcionar imágenes completas previamente reconstruidas, esto es, decodificadas (y correspondientes bloques y muestras de referencia) y/o una imagen actual parcialmente reconstruida (y correspondientes bloques y muestras 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 bloques 260, se configura para recibir u obtener el bloque de imagen 203 (un bloque de imagen actual 203 de la imagen actual 201) y datos de imagen reconstruida, por ejemplo, muestras de referencia de la misma imagen (actual) del búfer 216 y/o datos de imagen de referencia 231 de una o más imágenes decodificadas previamente del búfer de imágenes decodificadas 230, y 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 se puede configurar para seleccionar un modo de predicción (por ejemplo, un modo de intra- o interpredicción) y/o un correspondiente bloque de predicción 245 o 255 para ser usado como bloque de predicción 265, para cálculo del bloque residual 205 y para reconstrucción del bloque reconstruido 215.

En una realización, la unidad de selección de modo 262 se puede configurar 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 una mejor coincidencia o en otras palabras un residual mínimo (el residual mínimo significa mejor compresión para transmisión o almacenamiento), o proporciona mínimas sobrecargas de señalización (la mínimas sobrecargas de señalización significan mejor compresión para transmisión o almacenamiento), o considera o equilibra ambos. La unidad de selección de modo 262 se puede configurar para determinar el modo de predicción en función de optimización de distorsión de tasa (RDO), esto es, seleccionar un modo de predicción que proporciona una mínima distorsión de tasa o seleccionar un modo de predicción para el que distorsión de tasa relacionada satisface al menos un criterio de selección de modo de predicción.

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

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

Un conjunto de modos de intrapredicción puede comprender 35 diferentes modos de intrapredicción, por ejemplo, modos no direccionales tales como un modo DC (o promedio) y un modo plano, o modos direccionales tales como los definidos en H.265, o puede comprender 67 diferentes modos de intrapredicción, por ejemplo, modos no direccionales tales como un modo DC (o promedio) y un modo plano, o modos direccionales tales como los definidos en H.266 en desarrollo.

En una implementación posible, un conjunto de modos de interpredicción depende de imágenes de referencia disponibles (es decir, por ejemplo, imágenes decodificadas al menos parcialmente almacenadas en la DBP 230, como se ha descrito anteriormente) y otros parámetros de interpredicción, por ejemplo, depende de si una imagen de referencia entera o únicamente una parte de una imagen de referencia, por ejemplo, un área de ventana de búsqueda alrededor de un área de un bloque actual, se usa para buscar un bloque de referencia mejor coincidente y/o por ejemplo, depende de si se aplica interpolación de píxeles tal como interpolación de medio pel y/o de cuarto pel. El conjunto de modos de interpredicción puede comprender, por ejemplo, un modo de predicción de vector de movimiento avanzada (AMVP) y un modo de fusión. En una implementación específica, el conjunto de modos de interpredicción puede comprender un modo AMVP basado en punto de control refinado y un modo de fusión basado en punto de control refinado en las realizaciones de esta solicitud. En un ejemplo, la unidad de intrapredicción 254 se puede configurar para realizar cualquier combinación de tecnologías de interpredicción descritas más adelante.

Además de los modos de predicción anteriores, un modo omitir y/o un modo directo también pueden aplicarse en las realizaciones de esta solicitud.

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, usando iterativamente particionamiento de árbol cuádruple (QT), particionamiento de árbol binario (BT), particionamiento de árbol ternario (triple-árbol, 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 comprende selección de una estructura de árbol del bloque de imagen particionado 203 y la selección de un modo de predicción usado para cada una de las particiones de bloque o subbloques.

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

Por ejemplo, el codificador 20 se puede configurar para seleccionar un bloque de referencia de una pluralidad de bloques de referencia de una misma imagen o diferentes imágenes de una pluralidad de otras imágenes, y proporcionar una imagen de referencia y/o un desplazamiento (un desplazamiento espacial) entre una ubicación (coordenadas X, Y) del bloque de referencia y una ubicación del bloque actual como parámetro de interpredicción a la unidad de estimación de movimiento (que no se muestra en la FIGURA 2). Este desplazamiento también se llama vector de movimiento (MV).

La unidad de compensación de movimiento se configura para obtener un parámetro de interpredicción, y realizar 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 (que no se muestra en la FIGURA 2) puede comprender reunir o generar el bloque de predicción en función de un vector de movimiento/bloque determinado a través de estimación de movimiento (posiblemente realizando interpolación en precisión subpíxel). El filtrado de interpolación puede generar muestras de píxeles adicionales de muestras de píxeles conocidas. Esto potencialmente aumenta una cantidad de bloques de predicción candidatos que se pueden usar para codificar un bloque de imagen. Al recibir un vector de movimiento para un PU del bloque de imagen actual, la unidad de compensación de movimiento 246 puede localizar un bloque de predicción al que el vector de movimiento apunta en una de las listas de imágenes de referencia. La unidad de compensación de movimiento 246 también puede generar un elemento de sintaxis asociado con un bloque y un sector de vídeo, de modo que el decodificador 30 usa el elemento de sintaxis para decodificar el bloque de imagen en el sector de vídeo.

Específicamente, la unidad de interpredicción 244 puede transmitir un elemento de sintaxis a la unidad de codificado por entropía 270. El elemento de sintaxis comprende el parámetro de interpredicción (tal como información de indicación de selección de un modo de interpredicción usado para predicción del bloque actual tras atravesarse una pluralidad de modos de interpredicción). En un posible escenario de aplicación, si únicamente hay un modo de interpredicción, el parámetro de interpredicción puede no llevarse en el elemento de sintaxis. En este caso, el lado de decodificador 30 puede realizar directamente decodificación usando un modo de predicción predeterminado. Se puede entender que la unidad de interpredicción 244 se puede configurar para realizar cualquier combinación de tecnologías de interpredicción.

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

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

5 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, tras seleccionar un modo de intrapredicción para un bloque, la unidad de intrapredicción 254 se configura además para proporcionar un parámetro de intrapredicción, esto es, información que indica el modo de intrapredicción seleccionado para el bloque, a la unidad de codificado por entropía 270. En un ejemplo, la unidad de intrapredicción 254 se puede configurar para realizar cualquier combinación de tecnologías de intrapredicción.

15 Específicamente, la unidad de intrapredicción 254 puede transmitir el elemento de sintaxis a la unidad de codificado por entropía 270. El elemento de sintaxis comprende el parámetro de intrapredicción (tal como información de indicación de selección de un modo de intrapredicción usado para predicción del bloque actual tras atravesarse una pluralidad de modos de intrapredicción). En un posible escenario de aplicación, si únicamente hay un modo de intrapredicción, el parámetro de intrapredicción puede no llevarse en el elemento de sintaxis. En este caso, el lado de decodificador 30 puede realizar directamente decodificación usando un modo de predicción predeterminado.

20 La unidad de codificado por entropía 270 se configura para aplicar (o no aplicar) un algoritmo o esquema de codificado por entropía (por ejemplo, un esquema de codificado de longitud variable (VLC), un contexto-adaptativo VLC (CAVLC) esquema, un esquema de codificado aritmético, un esquema de codificado aritmético binario adaptativo por contexto (CABAC), un esquema de codificado aritmético binario adaptativo por contexto basado en sintaxis (SBAC), un esquema de codificado por entropía con particionamiento por intervalo de probabilidad (PIPE), u otra metodología o tecnología de codificado por entropía) a uno o todos del coeficiente residual cuantizado 209, el parámetro de interpredicción, el parámetro de intrapredicción y/o el parámetro de filtro de bucle, para obtener datos de imagen codificados 21 que pueden sacarse por medio de una salida 272, por ejemplo, en forma de flujo de bits codificado 21. El flujo de bits codificado puede transmitirse al decodificador de vídeo 30, o almacenarse para transmisión o recuperación subsiguiente por el decodificador de vídeo 30. La unidad de codificado por entropía 270 puede configurarse además para codificar entropía otro elemento de sintaxis para un sector de vídeo actual que se está codificando.

35 Otras variaciones estructurales del codificador de vídeo 20 se pueden usar para codificar un flujo de vídeo. Por ejemplo, el codificador basado en no transformada 20 puede cuantizar una señal residual directamente 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 combinadas en una sola unidad.

Específicamente, en esta realización de esta solicitud, el codificador 20 se puede configurar para implementar un método de interpredicción descrito en las siguientes realizaciones.

40 Debe entenderse que otras variaciones estructurales del codificador de vídeo 20 se pueden usar para codificar un flujo de vídeo. Por ejemplo, para algunos bloques de imágenes o fotogramas de imagen, el codificador de vídeo 20 puede cuantizar directamente una señal residual. En este caso, no se requiere el procesamiento por la unidad de procesamiento de transformada 206, y, correspondientemente, tampoco se requiere el procesamiento por la unidad de procesamiento de transformada inversa 212. Como alternativa, para algunos bloques de imágenes o fotogramas de imagen, el codificador de vídeo 20 no genera datos residuales. Correspondientemente, en este caso, 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. Como alternativa, el codificador de vídeo 20 puede almacenar directamente un bloque de imágenes reconstruido como bloque de referencia. En este caso, no se requiere el procesamiento por el filtro 220. Como alternativa, 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. Adicionalmente, en un caso de codificado por compresión sin pérdidas, 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 también son opcionales. Debe entenderse que, en diferentes escenarios de aplicación, la unidad de interpredicción 244 y la unidad de intrapredicción 254 se pueden usar selectivamente.

60 La FIGURA 3 es un diagrama de bloques esquemático/conceptual de un ejemplo de un decodificador 30 para implementar realizaciones de esta solicitud. El decodificador de vídeo 30 se configura para recibir, por ejemplo, datos de imagen codificados (por ejemplo, un flujo de bits codificado) 21 obtenidos a través de codificación por un codificador 20, para obtener una imagen decodificada 231. En un proceso de descodificación, el decodificador de vídeo 30 recibe, del codificador de vídeo 20, datos de vídeo, por ejemplo, un flujo de bits de vídeo codificado que representa un bloque de imagen en un sector de vídeo codificado y un elemento de sintaxis asociado.

65 En el ejemplo de la FIGURA 3, el codificador 30 comprende una unidad de unidad de decodificación de entropía 304, una unidad de cuantización inversa 310, una unidad de procesamiento de transformada inversa 312, una unidad de reconstrucción 314 (por ejemplo una sumador 314), un búfer 316, un filtro de bucle 320, un búfer de imágenes

5 decodificadas 330, y una unidad de procesamiento de predicción 360. La unidad de procesamiento de predicción 360 puede comprender 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 pasada de decodificación que generalmente es inversa a una pasada de codificación descrita con referencia al codificador de vídeo 20 en la FIGURA 2.

10 La unidad de decodificación por entropía 304 se configura para decodificar por entropía los datos de imagen codificados 21 para obtener, por ejemplo, un coeficiente cuantizado 309 y/o un parámetro de codificación decodificado (que no se muestra en la FIGURA 3), por ejemplo, cualquiera o todos de un parámetro de interpredicción, un parámetro de intrapredicción, un parámetro de filtro de bucle y/u otro elemento de sintaxis (que se decodifica). La unidad de decodificación por entropía 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 un elemento de sintaxis a nivel de sector de vídeo y/o nivel de bloque de imagen.

15 La unidad de cuantización inversa 310 puede ser idéntica en función a la unidad de cuantización inversa 110, la unidad de procesamiento de transformada inversa 312 puede ser idéntica en función a la unidad de procesamiento de transformada inversa 212, la unidad de reconstrucción 314 puede ser idéntica en función a la unidad de reconstrucción 214, el búfer 316 puede ser idéntico en función al búfer 216, el filtro de bucle 320 puede tener una función idéntica al filtro de bucle 220, y el búfer de imágenes decodificadas 330 puede ser idéntico en función al búfer de imágenes decodificadas 230.

20 La unidad de procesamiento de predicción 360 puede comprender la unidad de interpredicción 344 y la unidad de intrapredicción 354. La unidad de interpredicción 344 puede ser similar en función a la unidad de interpredicción 244, y la unidad de intrapredicción 354 puede ser similar en función a la unidad de intrapredicción 254. La unidad de procesamiento de predicción 360 usualmente se configura para realizar predicción de bloque y/u obtener un bloque de predicción 365 de los datos codificados 21, y recibir u obtener (explícitamente o implícitamente) un parámetro relacionado con predicción y/o información acerca de un modo de predicción seleccionado, por ejemplo, de la unidad de decodificación por entropía 304.

30 Cuando el sector de vídeo se codifica en un sector intracodificada (I), la unidad de intrapredicción 354 en la unidad de procesamiento de predicción 360 se configura para generar un bloque de predicción 365 de un bloque de imagen en el sector de vídeo actual en función de un modo de intrapredicción señalado y datos de un bloque decodificado previamente de un fotograma o imagen actual. Cuando el fotograma de vídeo se codifica en un sector intercodificada (es decir, B o P), la unidad de interpredicción 344 (por ejemplo, una unidad de compensación de movimiento) en la unidad de procesamiento de predicción 360 se configura para generar un bloque de predicción 365 de un bloque de vídeo en el sector de vídeo actual en función de un vector de movimiento y el otro elemento de sintaxis esto es recibido de la unidad de decodificación por entropía 304. En interpredicción, un bloque de predicción puede ser generado de una 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 y en función de imágenes de referencia almacenadas en la DPB 330.

45 La unidad de procesamiento de predicción 360 se configura para determinar información de predicción del bloque de vídeo en el sector de vídeo actual al analizar sintácticamente el vector de movimiento y el otro elemento de sintaxis, y generar, usando la información de predicción, el bloque de predicción del bloque de vídeo actual esto está siendo decodificado. En un ejemplo de esta solicitud, la unidad de procesamiento de predicción 360 determina, usando algunos elementos de sintaxis recibidos, un modo de predicción (por ejemplo, intrapredicción o interpredicción) para codificar el bloque de vídeo en el sector de vídeo, un tipo de sector interpredicción (por ejemplo, un sector B, un sector P, o un sector GPB), información de construcción de una o más de las listas de imágenes de referencia para el sector, un vector de movimiento de cada bloque de vídeo intercodificado en el sector, un estado de interpredicción de cada bloque de vídeo intercodificado en el sector, y otra información, para decodificar el bloque de vídeo en el sector de vídeo actual. En otro ejemplo de esta divulgación, el elemento de sintaxis recibido por el decodificador de vídeo 30 del flujo de bits comprende un elemento 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 una cabecera de sector.

55 La unidad de cuantización inversa 310 se puede configurar para realizar cuantización inversa (es decir, descuantización) en un coeficiente de transformada cuantizado proporcionado en el flujo de bits y decodificado por la unidad de decodificación por entropía 304. El proceso de cuantización inversa puede comprender el uso de un parámetro de cuantización calculado por el codificador de vídeo 20 para cada bloque de vídeo en el sector de vídeo para determinar un grado de cuantización que debe aplicarse y, de manera semejante, un grado de cuantización inversa que debe aplicarse.

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

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

5 La unidad de filtro de bucle 320 (ya sea en 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 transiciones de píxeles o mejorar la calidad de vídeo. En un ejemplo, la unidad de filtro de bucle 320 se puede configurar para realizar cualquier combinación de tecnologías de filtrado descritas más adelante. La unidad de filtro de bucle 320 está pensada para
10 representar uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de desplazamiento adaptativo por muestra (SAO), u otros filtros, por ejemplo, un filtro bilateral, un filtro de bucle adaptativo (ALF), un filtro de agudización o suavización, o un filtro colaborativo. Aunque la unidad de filtro de bucle 320 se muestra como filtro en bucle en la FIGURA 3, en otra implementación, la unidad de filtro de bucle 320 puede implementarse como filtro posbucle.

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

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

Otras variaciones del decodificador de vídeo 30 se pueden usar para decodificar un flujo de bits comprimido. Por ejemplo, el decodificador 30 puede generar un flujo de vídeo de salida sin la unidad de filtro de bucle 320. Por ejemplo, el decodificador basado en no transformada 30 puede cuantizar en inversa una señal residual directamente sin la unidad
25 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 combinadas en una sola unidad.

30 Específicamente, en esta realización de esta solicitud, el decodificador 30 se puede configurar para implementar un método de interpredicción descrito en las siguientes realizaciones.

Debe entenderse que otras variaciones estructurales del decodificador de vídeo 30 se pueden usar para decodificar un flujo de bits de vídeo codificado. Por ejemplo, el decodificador de vídeo 30 puede generar un flujo de vídeo de salida sin procesamiento por el filtro 320. Como alternativa, para algunos bloques de imágenes o fotogramas de
35 imagen, la unidad de decodificación por entropía 304 en el decodificador de vídeo 30 no obtiene un coeficiente cuantizado a través de decodificación. Correspondientemente, en este caso, 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. Adicionalmente, en un caso de compresión sin pérdida, la unidad de cuantización inversa 310 y la unidad de procesamiento de transformada inversa 312 también son opcionales. Debe entenderse que, en diferentes escenarios de aplicación, la unidad de interpredicción y la unidad de intrapredicción se pueden usar selectivamente.

40 Debe entenderse que, en el codificador 20 y el decodificador 30 en esta solicitud, un resultado de procesamiento de una etapa puede procesarse además y entonces sacarse a una etapa siguiente. Por ejemplo, tras una etapa tal como filtrado de interpolación, derivación de vector de movimiento, o filtrado den bucle, una operación adicional, tal como corte o desplazamiento, se realiza en un resultado de procesamiento de la correspondiente etapa.

45 Por ejemplo, puede procesarse además un vector de movimiento que es de un punto de control de un bloque de imagen actual y esto se deriva en función de un vector de movimiento de un bloque codificado afín vecino. Esto no está limitado en esta solicitud. Por ejemplo, un valor del vector de movimiento queda constreñido para estar dentro de un intervalo de anchura de bits específico. Asumiendo que una anchura de bit permitida del vector de movimiento es bitDepth, el valor del vector de movimiento va de $-2^{(\text{bitDepth} - 1)}$ a $2^{(\text{bitDepth} - 1)} - 1$, donde el símbolo " \wedge " representa exponenciación. Si bitdepth es 16, el valor va de -32768 a 32767. Si bitDepth es 18, el valor va de -131072 a 131071. El valor del vector de movimiento puede ser constreñido en cualquiera de las siguientes dos maneras:

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

$$u_x = (v_x + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}}$$

$$v_x = (u_x \geq 2^{\text{bitDepth} - 1}) ? (u_x - 2^{\text{bitDepth}}) : u_x$$

$$u_y = (v_y + 2^{\text{bitDepth}}) \% 2^{\text{bitDepth}}$$

$$v_y = (u_y \geq 2^{\text{bitDepth} - 1}) ? (u_y - 2^{\text{bitDepth}}) : u_y$$

Por ejemplo, un valor de vx es -32769, y 32767 se deriva 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 desbordamiento se descarta en un bit más significativo. Por lo tanto, un valor de vx es 0111,1111,1111,1111, esto es, 32767. Este valor está en consonancia con el resultado derivado a través de procesamiento según las fórmulas.

Manera 2: Se realiza recorte en el vector de movimiento, y se usan las siguientes fórmulas:

$$v_x = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, v_x)$$

$$v_y = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, v_y)$$

En las fórmulas anteriores, Clip3 se define como recortar un valor de z a un intervalo [x, y].

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; & z < x \\ y & ; & z > y \\ z & ; & \text{de otro modo} \end{cases}$$

La FIGURA 4 es un diagrama estructural esquemático 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 esta solicitud. El dispositivo de codificado de vídeo 400 es adecuado para implementar las realizaciones descritas 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 FIGURA 1A) o un codificador de vídeo (por ejemplo, el codificador 20 en la FIGURA 1A). En otra realización, el dispositivo de codificado de vídeo 400 puede ser uno o más componentes del decodificador 30 en la FIGURA 1A o el codificador 20 en la FIGURA 1A.

El dispositivo de codificado de vídeo 400 comprende: puertos de entrada 410 y un receptor (Rx) 420 que se configuran para recibir datos; un procesador, una unidad lógica, o una unidad de procesamiento central (CPU) 430 esto se configura para procesar datos; un transmisor (Tx) 440 y puertos de salida 450 que se configuran para transmitir datos; y una memoria 460 configurada para almacenar datos. El dispositivo de codificado de vídeo 400 puede comprender además componentes óptico-a-eléctrico y componentes eléctrico-a-óptico (EO) que se acoplan a los puertos de entrada 410, el receptor 420, el transmisor 440, y los puertos de salida 450, para entrada de una señal óptica o eléctrica.

El procesador 430 se implementa por hardware y software. El procesador 430 puede implementarse como uno o más chips de CPU, núcleos (por ejemplo, un procesador multinúcleo), FPGA, ASIC y DSP. El procesador 430 se comunica con los puertos de entrada 410, el receptor 420, el transmisor 440, los puertos de salida 450 y la memoria 460. El procesador 430 comprende 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 un método de predicción de bloque croma proporcionado en las realizaciones de esta solicitud. Por ejemplo, el módulo de codificación/decodificación 470 implementa, procesos, o proporciona diversas operaciones de codificado. Por lo tanto, el módulo de codificación/decodificación 470 proporciona una sustancial mejora a una función del dispositivo de codificado de vídeo 400, y afecta a la transformación del dispositivo de codificado de vídeo 400 a un estado diferente. Como alternativa, el módulo de codificación/decodificación 470 se implementa como instrucción que se almacena en la memoria 460 y es ejecutada por el procesador 430.

La memoria 460 comprende uno o más discos, unidades de cinta y unidades de estado sólido y se puede usar como dispositivo de almacenamiento de datos de desbordamiento, para almacenar programas cuando tales programas se seleccionan para ejecución, y para almacenar una instrucción y datos que se leen durante ejecución de programa. La memoria 460 puede ser volátil y/o no volátil, y puede ser una memoria de solo lectura (ROM), una memoria de acceso aleatorio (RAM), una memoria accesible por dirección de contenido ternario (TCAM) y/o una memoria de acceso aleatorio estática (SRAM).

La FIGURA 5 es un diagrama de bloques simplificado de un aparato 500 que se puede usar como uno cualquiera o ambos del dispositivo fuente 12 y el dispositivo de destino 14 en la FIGURA 1A según una realización de ejemplo. El aparato 500 puede implementar las tecnologías de esta solicitud. En otras palabras, la FIGURA 5 es un diagrama de bloques esquemático de una implementación de un dispositivo de decodificación o un dispositivo de descodificación (denominado dispositivo de codificado 500 por abreviar) según una realización de esta solicitud. El dispositivo de codificado 500 puede comprender 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 una instrucción. El procesador se configura para ejecutar la instrucción almacenada en la memoria. La memoria del dispositivo de codificado almacena 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. Para evitar la repetición, en esta memoria no se describen detalles.

En esta realización de esta solicitud, el procesador 510 puede ser una unidad de procesamiento central ("CPU" por abreviar). Como alternativa, el procesador 510 puede ser otro procesador de finalidad general, un procesador de señales digitales (DSP), un circuito integrado de aplicación específica (ASIC), una matriz de puertas programables en campo (FPGA) u otro dispositivo lógico programable, una compuerta discreta o un dispositivo lógico de transistores, un componente de hardware discreto, o algo semejante. El procesador de finalidad general puede ser un microprocesador, cualquier procesador convencional o algo semejante.

La memoria 530 puede comprender 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 puede usarse como alternativa como memoria 530. La memoria 530 puede comprender código y datos 531 a los que accede el procesador 510 a través del bus 550. La memoria 530 puede comprender además un sistema operativo 533 y un programa de aplicación 535. El programa de aplicación 535 comprende al menos un programa que permite al procesador 510 realizar un método de codificación y decodificación de vídeo descrito en esta solicitud. Por ejemplo, los programas de aplicación 535 pueden comprender aplicaciones 1 a N. Las aplicaciones comprenden además una aplicación de codificación o decodificación de vídeo (denominada aplicación de codificado de vídeo por abreviar) que realiza el método de codificación o decodificación de vídeo descrito en esta solicitud.

El sistema de bus 550 puede no únicamente comprender un bus de datos, sino también puede comprender un bus de energía, un bus de control, un bus de señal de estado y similares. Sin embargo, para una descripción clara, diversos tipos de buses en la figura se marcan como sistema de bus 550.

Opcionalmente, el dispositivo de codificado 500 puede comprender 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 y una unidad sensible al tacto que puede funcionar para detectar una entrada táctil. La pantalla 570 se puede conectar al procesador 510 a través del bus 550.

A continuación se describen primero conceptos en esta solicitud.

1. Modo de interpredicción:

En HEVC, se usan dos modos de interpredicción: un modo de predicción de vector de movimiento avanzada (AMVP) y un modo de fusión.

En el modo AMVP, bloques codificados vecinos espaciales o temporales (denotados como bloques adyacentes) de un bloque actual se atraviesan primero; una lista de vectores de movimiento candidatos (que también se pueden denominar lista de candidatos de información de movimiento) se construye en función de información de movimiento de los bloques vecinos; y entonces un vector de movimiento óptimo se determina de la lista de vectores de movimiento candidatos en función de un coste de distorsión de tasa. Información de movimiento de candidato correspondiente a un coste de distorsión de tasa mínimo se usa como predictor de vector de movimiento (MVP) del bloque actual. Ambas ubicaciones de los bloques vecinos y un orden de recorrido a través del mismo son predefinidas. El coste de distorsión de tasa se calcula según la Fórmula (1), donde J representa el coste de distorsión de tasa (coste de RD), SAD es una suma de diferencias absolutas (SAD) entre valores de muestra originales y valores de muestra de predicción obtenidos a través de estimación de movimiento usando un predictor de vector de movimiento candidato, R representa una tasa de bits, y λ representa un multiplicador de Lagrange. Un lado de codificador transfiere un valor de índice del predictor de vector de movimiento seleccionado en la lista de vectores de movimiento candidatos y un valor de índice de fotograma de referencia a un lado de decodificador. Además, se realiza búsqueda de movimiento en un vecino centrado en el MVP, para obtener un vector de movimiento real del bloque actual. El lado de codificador transfiere una diferencia (diferencia de vector de movimiento) entre el MVP y el vector de movimiento real al lado de decodificador.

$$J = SAD + \lambda R \tag{1}$$

En el modo de fusión, una lista de vectores de movimiento candidatos construye primero en función de información de movimiento de bloques codificados vecinos espaciales o temporales de un bloque actual. Entonces, se determina información de movimiento óptimo de la lista de vectores de movimiento candidatos como información de movimiento del bloque actual en función de un coste de distorsión de tasa. Un valor de índice (denotado como índice de fusión más adelante en esta memoria) de una ubicación de la información de movimiento óptimo en la lista de vectores de movimiento candidatos se transfiere a un lado de decodificador. Información de movimiento candidato espacial y temporal del bloque actual se muestra en la FIGURA 6A. La información de movimiento candidato espacial es de cinco bloques vecinos espaciales (A0, A1, B0, B1 y B2). Si un bloque vecino no está disponible (el bloque vecino no existe, o el bloque vecino no está codificado, o un modo de predicción usado para el bloque vecino no es un modo de interpredicción), información de movimiento del bloque vecino no se añade a la lista de vectores de movimiento candidatos. La información de movimiento candidato temporal del bloque actual se obtiene escalando un MV de un bloque en una ubicación correspondiente en un fotograma de referencia en función de recuento de orden de imágenes (picture order count, POC) del fotograma de referencia y un fotograma actual. Si un bloque en una ubicación T en el

fotograma de referencia está disponible se determina primero. Si el bloque no está disponible, se selecciona un bloque en una ubicación C.

5 Similar al modo AMVP, en el modo de fusión, ambas ubicaciones de los bloques vecinos y un orden de recorrido a través del mismo también se predefinen. Adicionalmente, las ubicaciones de los bloques vecinos y el orden transversal del mismo pueden ser diferentes en diferentes modos.

10 Se puede aprender que una lista de vectores de movimiento candidatos necesita mantenerse en ambos del modo AMVP y el modo de fusión. Antes de añadir nueva información de movimiento a la lista de candidatos cada vez, primero se comprueba si la misma información de movimiento existe en la lista. Si la misma información de movimiento existe en la lista, la información de movimiento no se añade a la lista. Este proceso de comprobación se denomina poda de la lista de vectores de movimiento candidatos. La poda de la lista es para evitar la existencia de la misma información de movimiento en la lista, para evitar cálculo redundante de coste de distorsión de tasa.

15 En interpredicción en HEVC, la misma información de movimiento se usa para todas las muestras de un bloque de codificado, y entonces se realiza compensación de movimiento en función de la información de movimiento, para obtener predictores de las muestras del bloque de codificado. En el bloque de codificado, sin embargo, no todas las muestras tienen los mismos rasgos de movimiento. El uso de la misma información de movimiento puede resultar en predicción imprecisa de compensación de movimiento y más información residual.

20 En estándares existentes de codificado de vídeo, se usa estimación de movimiento coincidente basado en modelo de movimiento traslacional, y se asume que el movimiento de todas las muestras en un bloque es congruente. Sin embargo, en el mundo real, hay una variedad de movimiento. Muchos objetos son en movimiento no traslacional, por ejemplo, un objeto rotatorio, un giro tipo montaña rusa en diferente direcciones, una pantalla de fuegos artificiales, y escenas peligrosas en películas, especialmente objeto en movimiento en un escenario de UGC. Para estos objetos en movimiento, si se usa una tecnología de compensación de movimiento de bloques basada en modelo de movimiento traslacional en los estándares de codificado existente para codificar, la eficiencia de codificado puede verse afectada enormemente. En vista de esto, un modelo de movimiento no traslacional, por ejemplo, un modelo de movimiento afín, se introduce para mejorar la eficiencia de codificado.

30 En función de esto, desde el punto de vista de diferentes modelos de movimiento, el modo AMVP puede clasificarse en un modo AMVP basado en modelo traslacional y un modo AMVP basado en modelo no traslacional, y el modo de fusión puede clasificarse en un modo de fusión basado en modelo traslacional y un modo de fusión basado en modelo no traslacional.

35 2. Modelo de movimiento no traslacional:

40 Predicción basada en modelo de movimiento no traslacional significa que se usa un mismo modelo de movimiento en ambos lados de codificador y de decodificador para derivar información de movimiento de cada subbloque de un bloque actual, y se realiza compensación de movimiento en función de la información de movimiento del subbloque para obtener un bloque de predicción. De esta manera, se mejora la eficiencia de predicción eficiencia. Modelos comunes de movimiento no traslacional comprenden un modelo de movimiento afín de 4 parámetros y un modelo de movimiento afín de 6 parámetros.

45 Un subbloque en las realizaciones de esta solicitud puede ser una muestra o un bloque de muestras $N_1 \times N_2$ obtenido usando un método de particionamiento particular, donde ambos N_1 y N_2 son enteros positivos, y N_1 puede ser igual a N_2 o puede no ser igual a N_2 .

50 El modelo de movimiento afín de 4 parámetros se expresa según la Fórmula (2):

$$\begin{cases} vx = a_1 + a_3x + a_4y \\ vy = a_2 - a_4x + a_3y \end{cases} \quad (2)$$

55 El modelo de movimiento afín de 4 parámetros puede ser representado por vectores de movimiento de dos muestras y sus coordenadas respecto a la muestra superior izquierda del bloque actual. Una muestra usada para representar un parámetro de modelo de movimiento se denomina punto de control. Si la muestra superior izquierda (0, 0) y la muestra superior derecha (W, 0) se usan como puntos de control, primero se determinan respectivos vectores de movimiento (vx_0, vy_0) y (vx_1, vy_1) del vértice superior izquierdo y el vértice superior derecho del bloque actual. Entonces, información de movimiento de cada subbloque del bloque actual se obtiene según la Fórmula (3), donde (x, y) representa coordenadas del subbloque respecto a la muestra superior izquierda del bloque actual, y W representa la anchura del bloque actual.

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W} x - \frac{vy_1 - vy_0}{W} y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W} x + \frac{vx_1 - vx_0}{W} y + vy_0 \end{cases} \quad (3)$$

El modelo de movimiento afín de 6 parámetros se expresa según la Fórmula (4):

$$\begin{cases} vx = a_1 + a_3x + a_4y \\ vy = a_2 + a_5x + a_6y \end{cases} \quad (4)$$

5 El modelo de movimiento afín de 6 parámetros puede ser representado por vectores de movimiento de tres muestras y su coordenadas respecto a la muestra superior izquierda del bloque actual. Si la muestra superior izquierda (0, 0), la muestra superior derecha (W, 0), y la muestra inferior izquierda (0, H) del bloque actual se usan como puntos de control, primero se determinan respectivos vectores de movimiento (vx₀, vy₀), (vx₁, vy₁), y (vx₂, vy₂) del punto de control superior izquierdo, el punto de control superior derecho, y el punto de control inferior izquierdo del bloque actual. 10 Entonces, información de movimiento de cada subbloque del bloque actual se obtiene según la Fórmula (5), donde (x, y) representa coordenadas del subbloque respecto a la muestra superior izquierda del bloque actual, y W y H representan la anchura y la altura del bloque actual, respectivamente.

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W}x + \frac{vx_2 - vx_0}{H}y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W}x + \frac{vy_2 - vy_0}{H}y + vy_0 \end{cases} \quad (5)$$

15 Un bloque de codificado que se predice usando un modelo de movimiento afín se denomina bloque codificado afín.

Generalmente, información de movimiento de un punto de control de un bloque codificado afín puede obtenerse usando un modo de predicción de vector de movimiento avanzada (AMVP) basado en modelo de movimiento afín o un afín modo de fusión basado en modelo de movimiento. 20

La información de movimiento del punto de control del bloque de codificado actual puede obtenerse usando un método de predicción de vector de movimiento de punto de control heredado o un método de predicción de vector de movimiento de punto de control construido. 25

3. Método de predicción de vector de movimiento de punto de control heredado:

El método de predicción de vector de movimiento de punto de control heredado es para usar un modelo de movimiento de un bloque codificado afín codificado vecino para determinar vectores de movimiento de punto de control candidato de un bloque actual. 30

Un bloque actual mostrado en la FIGURA 6B se usa como ejemplo. Bloques en ubicaciones vecinas alrededor del bloque actual se atraviesan en un orden especificado, por ejemplo, A1->B1->B0->A0->B2, para encontrar un bloque codificado afín que incluye un bloque en una ubicación vecina del bloque actual, y para obtener información de movimiento de punto de control del bloque codificado afín. Además, un vector de movimiento de punto de control (usado en el modo de fusión) o un predictor de vector de movimiento de punto de control (usado en el modo AMVP) del bloque actual se deriva usando un modelo de movimiento construido en función de la información de movimiento de punto de control del bloque codificado afín. El orden A1->B1->B0->A0->B2 se usa meramente como ejemplo. En esta solicitud también se puede usar otro orden de combinación. Adicionalmente, los bloques en las ubicaciones vecinas no se limitan a A1, B1, B0, A0, y B2. 35 40

El bloque en la ubicación vecina puede ser una muestra o un bloque de muestra de un tamaño preestablecido obtenido usando un método de particionamiento particular. Por ejemplo, el bloque de muestras puede ser un bloque de muestras 4x4, un bloque de muestras 4x2, o un bloque de muestra de otro tamaño. Esto no está limitado en esta memoria. 45

A continuación se describe un proceso de determinación usando A1 como ejemplo, y se pueden emplear procesos similares para otros casos.

Como se muestra en la FIGURA 4, si un bloque de codificado en el que se ubica A1 es un bloque codificado afín de 4 parámetros, se obtiene un vector de movimiento (vx₄, vy₄) del vértice superior izquierdo (x₄, y₄) y un vector de movimiento (vx₅, vy₅) del vértice superior derecho (x₅, y₅) del bloque codificado afín. Un vector de movimiento (vx₀, vy₀) del vértice superior izquierdo (x₀, y₀) del bloque actual codificado afín se calcula según la Fórmula (6), y un vector de movimiento (vx₁, vy₁) del vértice superior derecho (x₁, y₁) del bloque actual codificado afín se calcula según la Fórmula (7). 50 55

$$\begin{cases} vx_0 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_0 - x_4) - \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (y_0 - y_4) \\ vy_0 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_0 - x_4) + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (y_0 - y_4) \end{cases} \quad (6)$$

$$\begin{cases} vx_1 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_1 - x_4) - \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (y_1 - y_4) \\ vy_1 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_1 - x_4) + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (y_1 - y_4) \end{cases} \quad (7)$$

5 Una combinación del vector de movimiento (vx_0, vy_0) del vértice superior izquierdo (x_0, y_0) y el vector de movimiento (vx_1, vy_1) del vértice superior derecho (x_1, y_1) del bloque actual que se obtienen en función del bloque codificado afín en el que A1 se ubica es los vectores de movimiento candidatos de punto de control del bloque actual.

10 Si un bloque de codificado en el que A1 se ubica es un bloque codificado afín de 6 parámetros, se obtiene un vector de movimiento (vx_4, vy_4) del vértice superior izquierdo (x_1, y_4) , un vector de movimiento (vx_5, vy_5) del vértice superior derecho (x_5, y_5) , y un vector de movimiento (vx_6, vy_6) del vértice inferior izquierdo (x_6, y_6) del bloque codificado afín. Un vector de movimiento (vx_0, vy_0) del vértice superior izquierdo (x_0, y_0) del bloque actual se calcula según la Fórmula (8), un vector de movimiento (vx_1, vy_1) del vértice superior derecho (x_1, y_1) del bloque actual se calcula según la Fórmula (9), y un vector de movimiento (vx_2, vy_2) del vértice inferior izquierdo (x_2, y_2) del bloque actual se calcula según la Fórmula (10).

$$\begin{cases} vx_0 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_0 - x_4) + \frac{(vx_6 - vx_4)}{y_6 - y_4} \times (y_0 - y_4) \\ vy_0 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_0 - x_4) + \frac{(vy_6 - vy_4)}{y_6 - y_4} \times (y_0 - y_4) \end{cases} \quad (8)$$

$$\begin{cases} vx_1 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_1 - x_4) + \frac{(vx_6 - vx_4)}{y_6 - y_4} \times (y_1 - y_4) \\ vy_1 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_1 - x_4) + \frac{(vy_6 - vy_4)}{y_6 - y_4} \times (y_1 - y_4) \end{cases} \quad (9)$$

$$\begin{cases} vx_2 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_2 - x_4) + \frac{(vx_6 - vx_4)}{y_6 - y_4} \times (y_2 - y_4) \\ vy_2 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_2 - x_4) + \frac{(vy_6 - vy_4)}{y_6 - y_4} \times (y_2 - y_4) \end{cases} \quad (10)$$

20 Una combinación del vector de movimiento (vx_0, vy_0) del vértice superior izquierdo (x_0, y_0) , el vector de movimiento (vx_1, vy_1) del vértice superior derecho (x_1, y_1) , y el vector de movimiento (vx_2, vy_2) del vértice inferior izquierdo (x_2, y_2) del bloque actual que se obtienen en función del bloque codificado afín en el que se ubica A1 es el vector de movimiento candidato de punto de control del bloque actual.

25 Se debe observar que otros modelos de movimiento, ubicaciones candidatas y búsqueda y órdenes de recorrido a través también se pueden usar en esta solicitud. No se describen detalles en las realizaciones de esta solicitud.

30 Se debe observar que métodos para representar modelos de movimiento de bloques de codificado vecinos y actuales en función de otros puntos de control también se pueden usar en esta solicitud. En esta memoria no se describen detalles.

35 4. Método de predicción de vector de movimiento de punto de control construido (constructed control point motion vectors) 1:

El método de predicción de vector de movimiento de punto de control construido es para combinar vectores de movimiento de bloques codificados vecinos alrededor de un punto de control de un bloque actual como vector de movimiento de punto de control de un bloque actual codificado afín, sin considerar si los bloques codificados vecinos son bloques codificados afines.

40 Vectores de movimiento del vértice superior izquierdo y el vértice superior derecho del bloque actual se determinan en función de información de movimiento de los bloques codificados vecinos alrededor del bloque de codificado actual. La FIGURA 6C se usa como ejemplo para describir el método de predicción de vector de movimiento de punto de control construido. Se debe observar que la FIGURA 6C se usa meramente como ejemplo.

45 Como se muestra en la FIGURA 6C, vectores de movimiento de bloques codificados vecinos A2, B2, y B3 en la esquina superior izquierda se usan como vectores de movimiento candidatos para un vector de movimiento del vértice superior izquierdo del bloque actual; y vectores de movimiento de bloques codificados vecinos B1 y B0 en la parte esquina superior derecha se usan como vectores de movimiento candidatos para un vector de movimiento del vértice superior derecho del bloque actual. Los vectores de movimiento candidatos del vértice superior izquierdo y el vértice superior derecho se combinan para constituir una pluralidad de 2-tuplas. Vectores de movimiento de dos bloques codificados comprendidos en un 2-tupla se pueden usar como vectores de movimiento candidatos de punto de control del bloque actual, como se muestra en el siguiente Fórmula (11A):

$$\{v_{A2}, v_{B1}\}, \{v_{A2}, v_{B0}\}, \{v_{B2}, v_{B1}\}, \{v_{B2}, v_{B0}\}, \{v_{B3}, v_{B1}\}, \{v_{B3}, v_{B0}\} \quad (11A)$$

donde v_{A2} representa el vector de movimiento de A2, v_{B1} representa el vector de movimiento de B1, v_{B0} representa el vector de movimiento de B0, v_{B2} representa el vector de movimiento de B2, y v_{B3} representa el vector de movimiento de B3.

5 Como se muestra en la FIGURA 6C, vectores de movimiento de bloques codificados vecinos A2, B2, y B3 en la esquina superior izquierda se usan como vectores de movimiento candidatos para un vector de movimiento del vértice superior izquierdo del bloque actual; vectores de movimiento de bloques codificados vecinos B1 y B0 en la esquina superior derecha se usan como vectores de movimiento candidatos para un vector de movimiento del vértice superior derecho del bloque actual; y vectores de movimiento de bloques codificados vecinos A0 y A1 en la esquina inferior izquierda se usan como vectores de movimiento candidatos para un vector de movimiento del vértice inferior izquierdo del bloque actual. Los vectores de movimiento candidatos del vértice superior izquierdo, el vértice superior derecho, y el vértice inferior izquierdo se combinan para constituir 3-tuplas. Vectores de movimiento de tres bloques codificados comprendidos en un 3-tupla se pueden usar como vectores de movimiento candidatos de punto de control del bloque actual, como se muestra en el siguientes Fórmulas (11B) y (11C):

$$\{v_{A2}, v_{B1}, v_{A0}\}, \{v_{A2}, v_{B0}, v_{A0}\}, \{v_{B2}, v_{B1}, v_{A0}\}, \{v_{B2}, v_{B0}, v_{A0}\}, \{v_{B3}, v_{B1}, v_{A0}\}, \{v_{B3}, v_{B0}, v_{A0}\} \quad (11B)$$

$$\{v_{A2}, v_{B1}, v_{A1}\}, \{v_{A2}, v_{B0}, v_{A1}\}, \{v_{B2}, v_{B1}, v_{A1}\}, \{v_{B2}, v_{B0}, v_{A1}\}, \{v_{B3}, v_{B1}, v_{A1}\}, \{v_{B3}, v_{B0}, v_{A1}\} \quad (11C)$$

20 donde v_{A2} representa el vector de movimiento de A2, v_{B1} representa el vector de movimiento de B1, v_{B0} representa el vector de movimiento de B0, v_{B2} representa el vector de movimiento de B2, v_{B3} representa el vector de movimiento de B3, v_{A0} representa el vector de movimiento de A0, y v_{A1} representa el vector de movimiento de A1.

25 Se debe observar que otros métodos para combinar vectores de movimiento de punto de control también se pueden usar en esta solicitud. En esta memoria no se describen detalles.

30 Se debe observar que métodos para representar modelos de movimiento de bloques de codificado vecinos y actuales en función de otros puntos de control también se pueden usar en esta solicitud. En esta memoria no se describen detalles.

5. Método de predicción de vector de movimiento de punto de control construido (constructed control point motion vectors) 2, como se muestra en la FIGURA 6D:

35 Etapa 601: Obtener información de movimiento de puntos de control de un bloque actual.

Por ejemplo, en la FIGURA 6C, CPk (k = 1, 2, 3, 4) representa un k-ésimo punto de control. A0, A1, A2, B0, B1, B2 y B3 son ubicaciones vecinas espaciales del bloque actual y se usan para predecir CP1, CP2, o CP3. T es una ubicación vecina temporal del bloque actual y se usa para predecir CP4.

40 Se asume que las coordenadas de CP1, CP2, CP3 y CP4 son (0, 0), (W, 0), (H, 0), y (W, H), respectivamente, donde W y H representan la anchura y la altura del bloque actual.

45 Para cada punto de control, información de movimiento del punto de control se obtiene en el siguiente orden:

(1) Para CP1, un orden de comprobación es B2->A2->B3. Si B2 está disponible, se usa información de movimiento de B2 para CP1. De otro modo, se comprueba A2 y B3. Si no está disponible información de movimiento de todas las tres ubicaciones, no se puede obtener información de movimiento de CP1.

50 (2) Para CP2, un orden de comprobación es B0->B1. Si B0 está disponible, se usa información de movimiento de B0 para CP2. De otro modo, se comprueba B1. Si no está disponible información de movimiento de ambos de las ubicaciones, no se puede obtener información de movimiento de CP2.

(3) Para CP3, un orden de comprobación es A0->A1.

55 (4) Para CP4, se usa información de movimiento de T.

60 En esta memoria, que X está disponible significa que un bloque en X (X es A0, A1, A2, B0, B1, B2, B3 o T) ya está codificado y se usa un modo de interpredicción para el bloque. De lo contrario, X no está disponible.

Se debe observar que otros métodos para obtener información de movimiento de punto de control también se pueden usar en esta solicitud. En esta memoria no se describen detalles.

65 Etapa 602: Combinar la información de movimiento de los puntos de control para obtener información de movimiento de punto de control construido.

Información de movimiento de dos puntos de control se combina para constituir un 2-tupla, para construir un modelo de movimiento afín de 4 parámetros. Combinaciones de dos puntos de control pueden ser {CP1, CP4}, {CP2, CP3}, {CP1, CP2}, {CP2, CP4}, {CP1, CP3}, y {CP3, CP4}. Por ejemplo, un modelo de movimiento afín de 4 parámetros construido usando un 2-tupla que incluye los puntos de control CP1 y CP2 puede denotarse como Afín (CP1, CP2).

Información de movimiento de tres puntos de control se combina para constituir un 3-tupla, para construir un modelo de movimiento afín de 6 parámetros. Combinaciones de tres puntos de control pueden ser {CP1, CP2, CP4}, {CP1, CP2, CP3}, {CP2, CP3, CP4}, y {CP1, CP3, CP4}. Por ejemplo, un modelo de movimiento afín de 6 parámetros construido usando un 3-tupla que incluye los puntos de control CP1, CP2, y CP3 puede denotarse como Afín (CP1, CP2, CP3).

Información de movimiento de cuatro puntos de control se combina para constituir un 4-tupla, para construir un modelo de movimiento bilineal de 8 parámetros. Un modelo bilineal de 8 parámetros construido usando un 4-tupla que incluye los puntos de control CP1, CP2, CP3, y CP4 puede denotarse como Bilineal (CP1, CP2, CP3, CP4).

En las realizaciones de esta solicitud, para facilitar la descripción, una combinación de información de movimiento de dos puntos de control (o dos bloques codificados) simplemente se denomina un 2-tupla, una combinación de información de movimiento de tres puntos de control (o tres bloques codificados) simplemente se denomina un 3-tupla, y una combinación de información de movimiento de cuatro puntos de control (o cuatro bloques codificados) simplemente se denomina un 4-tupla.

Estos modelos se atraviesan en un orden preestablecido. Si información de movimiento de un punto de control correspondiente a una combinación modelo no está disponible, se considera que el modelo no está disponible. De otro modo, se determina un índice de fotograma de referencia del modelo, y se escala un vector de movimiento del punto de control. Si información de movimiento de todas los puntos de control tras escalar es congruente, el modelo es inválido. Si toda la información de movimiento de puntos de control que controla el modelo está disponible, y el modelo es válido, la información de movimiento de los puntos de control usada para construir el modelo se añade a una lista de candidatos de información de movimiento.

Un método de escalado de vector de movimiento de punto de control se muestra en la Fórmula (12):

$$MV_s = \frac{CurPoc - DesPoc}{CurPoc - SrcPoc} \times MV \quad (12)$$

donde *CurPoc* representa un número POC de un fotograma actual, *DesPoc* representa un número POC de un fotograma de referencia de un bloque actual, *SrcPoc* representa un número POC de un fotograma de referencia de un punto de control, MV_s representa un vector de movimiento obtenido tras escalado, y *MV* representa un vector de movimiento de un punto de control.

Se debe observar que diferentes combinaciones de puntos de control pueden convertirse en puntos de control en una misma ubicación.

Por ejemplo, un modelo de movimiento afín de 4 parámetros obtenido en función de una combinación de {CP 1, CP4}, {CP2, CP3}, {CP2, CP4}, {CP1, CP3}, o {CP3, CP4} se representa por {CP1, CP2} o {CP1, CP2, CP3} tras la conversión. Un método de conversión comprende: sustituir vectores de movimiento e información de coordenadas de puntos de control en la Fórmula (2) para obtener parámetros de modelo; y entonces sustituir información de coordenadas de {CP1, CP2} en la Fórmula (3) para obtener vectores de movimiento de {CP1, CP2}.

Más directamente, puede realizarse conversión según las fórmulas (13) a (21), donde *W* representa la anchura del bloque actual, y *H* representa la altura del bloque actual. En las Fórmulas (13) a (21), (vx_0, vy_0) representa un vector de movimiento de CP1, (vx_1, vy_1) representa un vector de movimiento de CP2, (vx_2, vy_2) representa un vector de movimiento de CP3, y (vx_3, vy_3) representa un vector de movimiento de CP4.

{CP1, CP2} puede convertirse en {CP1, CP2, CP3} según la Fórmula (13). En otras palabras, el vector de movimiento de CP3 en {CP1, CP2, CP3} se puede determinar según la Fórmula (13):

$$\begin{cases} vx_2 = -\frac{vy_1 - vy_0}{W} H + vx_0 \\ vy_2 = +\frac{vx_1 - vx_0}{W} H + vy_0 \end{cases} \quad (13)$$

{CP1, CP3} puede convertirse a {CP1, CP2} o {CP1, CP2, CP3} según la Fórmula (14):

$$\begin{cases} vx_1 = +\frac{vy_2 - vy_0}{H} W + vx_0 \\ vy_1 = -\frac{vx_2 - vx_0}{H} W + vy_0 \end{cases} \quad (14)$$

{CP2, CP3} puede convertirse en {CP1, CP2} o {CP1, CP2, CP3} según la Fórmula (15):

$$\begin{cases} vx_0 = \frac{vx_2 - vx_1}{W+W+H+H} W * W - \frac{vy_2 - vy_1}{W+W+H+H} H * W + vx_1 \\ vy_0 = \frac{vy_2 - vy_1}{W+W+H+H} W * W + \frac{vx_2 - vx_1}{W+W+H+H} H * W + vy_1 \end{cases} \quad (15)$$

5

{CP1, CP4} puede convertirse en {CP1, CP2} o {CP1, CP2, CP3} según la Fórmula (16) o (17):

$$\begin{cases} vx_1 = \frac{vx_3 - vx_0}{W+W+H+H} W * W + \frac{vy_3 - vy_0}{W+W+H+H} H * W + vx_0 \\ vy_1 = \frac{vy_3 - vy_0}{W+W+H+H} W * W - \frac{vx_3 - vx_0}{W+W+H+H} H * W + vy_0 \end{cases} \quad (16)$$

$$\begin{cases} vx_2 = \frac{vx_3 - vx_0}{W+W+H+H} H * H - \frac{vy_3 - vy_0}{W+W+H+H} H * W + vx_0 \\ vy_2 = \frac{vy_3 - vy_0}{W+W+H+H} W * H + \frac{vx_3 - vx_0}{W+W+H+H} H * H + vy_0 \end{cases} \quad (17)$$

10

{CP2, CP4} puede convertirse en {CP1, CP2} según la Fórmula (18), y {CP2, CP4} puede convertirse en {CP1, CP2, CP3} según las fórmulas (18) y (19):

$$\begin{cases} vx_0 = -\frac{vy_3 - vy_1}{H} W + vx_1 \\ vy_0 = +\frac{vx_3 - vx_1}{H} W + vy_1 \end{cases} \quad (18)$$

15

$$\begin{cases} vx_2 = -\frac{vy_3 - vy_1}{H} W + vx_3 \\ vy_2 = +\frac{vx_3 - vx_1}{H} W + vy_3 \end{cases} \quad (19)$$

{CP3, CP4} puede convertirse en {CP1, CP2} según la Fórmula (20), y {CP3, CP4} puede convertirse en {CP1, CP2, CP3} según las fórmulas (20) y (21):

20

$$\begin{cases} vx_0 = +\frac{vy_3 - vy_2}{W} H + vx_2 \\ vy_0 = -\frac{vx_3 - vx_2}{W} H + vy_2 \end{cases} \quad (20)$$

$$\begin{cases} vx_1 = +\frac{vy_3 - vy_2}{W} H + vx_3 \\ vy_1 = -\frac{vx_3 - vx_2}{W} H + vy_3 \end{cases} \quad (21)$$

25

Por ejemplo, un modelo de movimiento afín de 6 parámetros obtenido en función de una combinación {CP1, CP2, CP4}, {CP2, CP3, CP4}, o {CP1, CP3, CP4} se convierte y representa usando {CP1, CP2, CP3}. Un método de conversión comprende: sustituir vectores de movimiento e información de coordenadas de puntos de control en la Fórmula (4) para obtener parámetros de modelo; y entonces sustituir información de coordenadas de {CP1, CP2, CP3} en la Fórmula (5) para obtener vectores de movimiento de {CP1, CP2, CP3}.

30

Más directamente, puede ser realizar conversión según las fórmulas (22) a (24), donde W representa la anchura del bloque actual, y H representa la altura del bloque actual. En las Fórmulas (13) a (21), (vx₀, vy₀) representa un vector de movimiento de CP1, (vx₁, vy₁) representa un vector de movimiento de CP2, (vx₂, vy₂) representa un vector de movimiento de CP3, y (vx₃, vy₃) representa un vector de movimiento de CP4.

35

{CP1, CP2, CP4} puede convertirse en {CP1, CP2, CP3} según la Fórmula (22):

$$\begin{cases} vx_2 = vx_3 + vx_0 - vx_1 \\ vy_2 = vy_3 + vy_0 - vy_1 \end{cases} \quad (22)$$

40

{CP2, CP3, CP4} puede convertirse en {CP1, CP2, CP3} según la Fórmula (23):

$$\begin{cases} vx_0 = vx_1 + vx_2 - vx_3 \\ vy_0 = vy_1 + vy_2 - vy_3 \end{cases} \quad (23)$$

45

{CP1, CP3, CP4} puede convertirse en {CP1, CP2, CP3} según la Fórmula (24):

$$\begin{cases} vx_1 = vx_3 + vx_0 - vx_2 \\ vy_1 = vy_3 + vy_0 - vy_2 \end{cases} \quad (24)$$

6. Predicción de vector de movimiento temporal avanzada (ATMVP):

En interpredicción en HEVC, se realiza compensación de movimiento en todas las muestras de un bloque actual en función de la misma información de movimiento para obtener predictores de las muestras del bloque a procesar. Sin embargo, no todas las muestras del bloque a procesar tienen los mismos rasgos de movimiento. El uso de la misma información de movimiento para predecir todas las muestras del bloque a procesar puede reducir la precisión de compensación de movimiento y aumentar la información residual.

Para resolver los problemas anteriores, en una solución existente se proporciona una tecnología de predicción de vector de movimiento temporal avanzada (ATMVP).

Un proceso para realizar predicción usando la tecnología ATMVP principalmente comprende las siguientes etapas, como se muestra en la FIGURA 6E:

- (1) Determinar un vector de desplazamiento del bloque a procesar.
- (2) Determinar, en un objetivo de imagen correspondiente en función de una ubicación de un subbloque a procesar del bloque a procesar y el vector de desplazamiento, un subbloque correspondiente al subbloque a procesar, donde el objetivo de imagen es una de imágenes codificadas.
- (3) Determinar un vector de movimiento del subbloque a procesar en función de un vector de movimiento del correspondiente subbloque.

Por ejemplo, el vector de movimiento del subbloque a procesar actual se puede determinar usando un método de escalado. Por ejemplo, el método de escalado se implementa según la Fórmula (25):

$$MV_c = \frac{CPoc - DPoc}{DPoc - SPoc} \times MV_g \quad (25)$$

donde CPoc representa un número POC de un fotograma en el que se ubica el bloque a procesar, DPoc representa un número POC de un fotograma en el que se ubica el correspondiente subbloque, SrcPoc representa un número POC de un fotograma de referencia del correspondiente subbloque, MV_c representa un vector de movimiento obtenido a través de escalado, y MV_g representa el vector de movimiento del correspondiente subbloque.

- (4) Realizar 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 predicción de muestra del subbloque a procesar.

7. Predicción de vector de movimiento plano (PLANO):

En predicción de vector de movimiento plano, se obtiene información de movimiento en una ubicación vecina espacial superior, una ubicación vecina espacial izquierda, una ubicación derecha y una ubicación inferior de cada subbloque a procesar del bloque a procesar, y se calcula un promedio de la información de movimiento, y se convierte en información de movimiento del subbloque a procesar actual.

Para un subbloque a procesar cuyas coordenadas son (x, y) , un vector de movimiento $P(x, y)$ del subbloque a procesar se calcula en función de un vector de movimiento de interpolación horizontal $P_h(x, y)$ y un vector de movimiento de interpolación horizontal $P_v(x, y)$ según la Fórmula (26):

$$P(x, y) = (H \times P_h(x, y) + W \times P_v(x, y) + H \times W) / (2 \times H \times W) \quad (26)$$

donde H representa la altura del bloque a procesar, y W representa la anchura del bloque a procesar.

El vector de movimiento de interpolación horizontal $P_h(x, y)$ y el vector de movimiento de interpolación horizontal $P_v(x, y)$ pueden calcularse en función de vectores de movimiento de subbloques en la izquierda, la derecha, la parte superior y la parte inferior del subbloque a procesar actual según las fórmulas (27) y (28):

$$P_h(x, y) = (W - 1 - x) \times L(-1, y) + (x + 1) \times R(w, y) \quad (27)$$

$$P_v(x, y) = (H - 1 - y) \times A(x, -1) + (y + 1) \times B(x, H) \quad (28)$$

donde $L(-1, y)$ representa un vector de movimiento de un subbloque en la izquierda del subbloque a procesar, $R(w, y)$ representa un vector de movimiento de un subbloque en la derecha del subbloque a procesar, $A(x, -1)$ representa un vector de movimiento de un subbloque en la parte superior del subbloque a procesar, y $B(x, H)$ representa un vector de movimiento de un subbloque en la parte inferior del subbloque a procesar.

Un vector de movimiento L de un bloque izquierdo y un vector de movimiento A de un bloque superior se obtienen en función de un bloque vecino espacial de un bloque de codificado actual. Vectores de movimiento L(-1, y) y A(x, -1) de bloques de codificado en ubicaciones preestablecidas (-1, y) y (x, -1) se obtienen en función de las coordenadas (x, y) del subbloque a procesar.

Como se muestra en la FIGURA 7, un vector de movimiento R(w, y) de un bloque derecho y un vector de movimiento B(x, H) de un bloque inferior puede extraerse usando el siguiente método:

(1) Extraer información de movimiento temporal BR en una ubicación vecina espacial inferior derecha del bloque a procesar.

(2) Obtener el vector de movimiento R(w, y) del bloque derecho al realizar cálculo de ponderación en función de vector de movimiento AR extraído en una ubicación vecina espacial superior derecha y la información de movimiento temporal extraída BR en la ubicación vecina espacial inferior derecha, como se muestra en la Fórmula (29):

$$R(w, y) = ((H - y - 1) * AR + (y + 1) * BR) / H \quad (29)$$

(3) Obtener el vector de movimiento B(x, H) del bloque inferior al realizar cálculo de ponderación en función de vector de movimiento extraído BL en una ubicación vecina espacial inferior izquierda y la información de movimiento temporal extraído BR en la ubicación vecina espacial inferior derecha, como se muestra en la Fórmula (30):

$$B(x, H) = ((W - x - 1) * BL + (x + 1) * BR) / H \quad (30)$$

Se debe observar que el vector de movimiento usado en el cálculo es un vector de movimiento obtenido a través de escalado tras escalar el vector de movimiento para apuntar al primer fotograma de referencia en una cola específica de fotogramas de referencia.

8. Modo de predicción de vector de movimiento avanzada basada en modelo de movimiento afín (modo AMVP afín):

(1) Construcción una lista de vectores de movimiento candidatos.

Una lista de vectores de movimiento candidatos correspondiente al modo AMVP basado en modelo de movimiento afín se construye usando el método de predicción de vector de movimiento de punto de control heredado y/o el método de predicción de vector de movimiento de punto de control construido. En las realizaciones de esta solicitud, la lista de vectores de movimiento candidatos correspondiente al modo AMVP basado en modelo de movimiento afín puede denominarse lista de candidatos de predictor de vector de movimiento de punto de control (control point motion vectors predictor candidate list). Un predictor de vector de movimiento de cada punto de control comprende vectores de movimiento de dos puntos de control (para un modelo de movimiento afín de 4 parámetros) o vectores de movimiento de tres puntos de control (para un modelo de movimiento afín de 6 parámetros).

Opcionalmente, la lista de candidatos de predictor de vector de movimiento de punto de control se poda y ordena según una regla particular, y puede truncarse o rellenarse a una cantidad particular.

(2) Determinar un predictor óptimo de vector de movimiento de punto de control.

En un lado de codificador, un vector de movimiento de cada subbloque de un bloque de codificado actual se obtiene en función de cada predictor de vector de movimiento de punto de control en la lista de candidatos de predictor de vector de movimiento de punto de control según la Fórmula (3) o (5). El vector de movimiento obtenido se usa para obtener un valor de muestra en una ubicación correspondiente en un fotograma de referencia al que apunta el vector de movimiento del subbloque. El valor de muestra obtenido se usa como predictor para realizar compensación de movimiento usando un modelo de movimiento afín. Se calcula una diferencia promedio entre un valor original y un valor de predicción de cada muestra del bloque de codificado actual. Un predictor de vector de movimiento de punto de control correspondiente a una diferencia promedio mínima se selecciona como el predictor óptimo de vector de movimiento de punto de control, y se usan predictores de vector de movimiento de dos o tres puntos de control del bloque de codificado actual. Un número de índice que representa una ubicación del predictor de vector de movimiento de punto de control en la lista de candidatos de predictor de vector de movimiento de punto de control se codifica en un flujo de bits y se envía a un decodificador.

En el lado de decodificador, el número de índice se analiza sintácticamente, y el predictor de vector de movimiento de punto de control (CPMVP) se determina de la lista de candidatos de predictor de vector de movimiento de punto de control en función del número de índice.

(3) Determinar un vector de movimiento de punto de control

En el lado de codificador, un predictor de vector de movimiento de punto de control se usa como punto de inicio de búsqueda para búsqueda de movimiento dentro de un intervalo de búsqueda específico, para obtener un vector de movimiento de punto de control (CPMV). Una diferencia entre el vector de movimiento de punto de control y el predictor de vector de movimiento de punto de control (CPMVD) se transfiere al lado de decodificador.

En el lado de decodificador, la diferencia de vector de movimiento de punto de control se analiza sintácticamente y se añade al predictor de vector de movimiento de punto de control, para obtener el vector de movimiento de punto de control.

9. Modo de fusión de subbloque (sub-block based merging mode):

Una lista de candidatos de fusión basada en subbloques (sub-block based merging candidate list) se construye usando al menos un de la predicción de vector de movimiento temporal avanzada, el método de predicción de vector de movimiento de punto de control heredado, el método de predicción de vector de movimiento de punto de control construido, y el método de predicción plano.

Opcionalmente, la lista de candidatos de fusión basada en subbloques se poda y ordena según una regla particular, y puede truncarse o rellenarse a una cantidad particular.

En un lado de codificador, si se usa la predicción de vector de movimiento temporal avanzada, un vector de movimiento de cada subbloque (una muestra o un bloque de muestras $N_1 \times N_2$ obtenido usando un método de particionamiento particular) se obtiene usando el método descrito en las descripciones anteriores en "7. Predicción de vector de movimiento plano". Si se usa el método de predicción plano, se obtiene un vector de movimiento de cada subbloque usando el método descrito en las descripciones anteriores en "8. Modo de predicción de vector de movimiento avanzada basado en modelo de movimiento afín".

Si se usa el método de predicción de vector de movimiento de punto de control heredado o el método de predicción de vector de movimiento de punto de control construido, un vector de movimiento de cada subbloque (una muestra o un bloque de muestras $N_1 \times N_2$ obtenido usando un método de particionamiento particular) de un bloque actual se obtiene según la Fórmula (3) o (5). Tras obtenerse el vector de movimiento de cada subbloque, un valor de muestra en una ubicación en un fotograma de referencia a que el vector de movimiento del subbloque puntos se obtiene además, y el valor de muestra se usa como predictor del subbloque para compensación de movimiento afín. Se calcula una diferencia promedio entre un valor original y un predictor de cada muestra de un bloque de codificado actual. Un vector de movimiento de punto de control correspondiente a una diferencia promedio mínima se selecciona como vectores de movimiento de dos o tres puntos de control del bloque de codificado actual. Un número de índice que representa una ubicación del vector de movimiento de punto de control en la lista de candidatos se codifica en un flujo de bits y se envía a un decodificador.

En el lado de decodificador, el número de índice se analiza sintácticamente, y el vector de movimiento de punto de control (vectores de movimiento de punto de control, CPMV) se determina de la lista de candidatos de fusión de vector de movimiento de punto de control en función del número de índice.

Además, se debe observar que en esta solicitud, "al menos uno" significa uno o más, y "una pluralidad de" significa dos o más de dos. El término "y/o" describe una relación de asociación para describir objetos asociados e representa que pueden existir tres relaciones. Por ejemplo, A y/o B pueden representar los siguientes casos: Únicamente A existe, ambos A y B existen, y únicamente B existe, donde A y B pueden ser singulares o plurales. El carácter "/" generalmente representa una relación "o" entre los objetos asociados. "Al menos un artículo (pieza) de los siguientes" o una expresión similar a la misma significa cualquier combinación de estos artículos, incluido un solo artículo (pieza) o cualquier combinación de una pluralidad de artículos (piezas). Por ejemplo, al menos un artículo (pieza) de a, b o c puede representar: a, b, c, a-b, a-c, b-c, o a-b-c, donde a, b, y c pueden ser singulares o plurales.

En esta solicitud, cuando se usa un modo de interpredicción para decodificar un bloque actual, un elemento de sintaxis se puede usar para señalar el modo de interpredicción.

Para algunas estructuras de sintaxis usadas actualmente del modo de interpredicción usado para analizar sintácticamente el bloque actual, consúltese la Tabla 1. Se debe observar que un elemento de sintaxis en una estructura de sintaxis puede como alternativa representarse por otros indicadores. Esto no se limita específicamente en esta solicitud.

Tabla 1

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
if(slice_type != I) {	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0)	
pred_mode_flag[x0][y0]	ae(v)
coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
si no	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode[x0][y0]	ae(v)
} else { /* MODE_INTER (inter mode) */	
if(cu_skip_flag[x0][y0]) {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 0 && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1 && MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} si no {	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if((sps_affine_enabled_flag sps_sbtmvp_enabled_flag) && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 0 && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1)	
merge_subblock_idx[x0][y0]	ae(v)
} si no {	
if(slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)
coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	

inter_affine_flag[x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(num_ref_idx_l0_active_minus1 > 0)	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag[x0][y0]	ae(v)
} si no {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} si no {	
mvd_coding(x0, y0, 1, 0)	
coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
mvp_l1_flag[x0][y0]	ae(v)
} si no {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	

if(sps_amvr_enabled_flag && inter_affine_flag == 0 &&	
(MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0	
MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0))	
amvr_mode[x0][y0]	ae(v)
}	
}	
}	
if(CuPredMode[x0][y0] != MODE_INTRA && cu_skip_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
}	

La variable treeType especifica un tipo de árbol de codificado usado para codificar un bloque actual.

- 5 La variable slice_type se usa para especificar un tipo de un sector en el que se ubica un bloque actual, por ejemplo, un tipo P, un tipo B, o un tipo I.

10 El elemento de sintaxis cu_skip_flag[x0][y0] se puede usar para especificar si un bloque actual tiene un residual. Por ejemplo, cuando cu_skip_flag[x0][y0] = 1, indica que el bloque actual tiene el residual; o cuando cu_skip_flag[x0][y0] = 0, indica que el bloque actual no tiene residual.

15 Se debe observar que un modo omitir es un modo de fusión especial. Tras usarse un modo de fusión para encontrar un vector de movimiento, si un codificador determina, usando un método, que el bloque actual es básicamente el mismo que el bloque de referencia, no es necesario transmitir datos residuales, y únicamente se tiene que transmitir un índice del vector de movimiento y un cu_skip_flag. Por lo tanto, si cu_skip_flag[x0][y0] = 0, indica que el bloque actual no tiene residual y no es necesario analizar sintácticamente datos residuales.

20 El elemento de sintaxis pred_mode_flag[x0][y0] se usa para especificar si un modo de predicción para un bloque actual es un modo de interpredicción o intrapredicción.

La variable CuPredMode [x0][y0] se determina en función de pre_mode_flag[x0][y0]. MODE_INTRA especifica un modo de intrapredicción.

25 El elemento de sintaxis merge_flag[x0][y0] se puede usar para especificar si un modo de fusión se usa para un bloque actual. Por ejemplo, cuando merge_flag[x0][y0] = 1, indica que el modo de fusión se usa para el bloque actual; o cuando merge_flag[x0][y0] = 0, indica que el modo de fusión no se usa para el bloque actual, donde x0 y y0 representan coordenadas del bloque actual en una imagen de vídeo.

30 cbWidth especifica la anchura del bloque actual, y cbHeight especifica la altura del bloque actual.

35 El elemento de sintaxis sps_affine_enabled_flag se usa para especificar si un modelo de movimiento afín se puede usar para interpredecir en un bloque de imagen comprendido en una imagen de vídeo. Por ejemplo, cuando sps_affine_enabled flag = 0, indica que el modelo de movimiento afín no se puede usar para interpredecir el bloque de imagen comprendido en la imagen de vídeo; o cuando sps_affine_enabled flag = 1, indica que el modelo de movimiento afín se puede usar para interpredecir el bloque de imagen comprendido en la imagen de vídeo.

40 El elemento de sintaxis merge_subblock_flag[x0][y0] se puede usar para especificar si un modo de fusión basado en subbloques se usa para un bloque actual. Un tipo (slice_type) de un sector en el que se ubica el bloque actual es un tipo P o un tipo B. Por ejemplo, cuando merge_subblock_flag[x0][y0] = 1, indica que el modo de fusión basado en subbloques se usa para el bloque actual; o cuando merge_subblock_flag[x0][y0] = 0, indica que el modo de fusión basado en subbloques no se usa para el bloque actual, pero se puede usar un modo de fusión basado en modelo de movimiento traslacional.

45 El elemento de sintaxis merge_idx[x0][y0] se puede usar para especificar un índice de una lista de candidatos de fusión.

El elemento de sintaxis `merge_subblock_idx[x0][y0]` se puede usar para especificar un índice de una lista de candidatos de fusión basada en subbloques.

5 `sps_sbtmvp_enabled_flag` se usa para especificar si un modo ATMVP se puede usar para interpredicir un bloque de imagen comprendido en una imagen de vídeo. Por ejemplo, cuando `sps_sbtmvp_enabled_flag = 1`, indica que el modo ATMVP se puede usar para interpredicir el bloque de imagen comprendido en la imagen de vídeo; o cuando `sps_sbtmvp_enabled_flag = 0`, indica que el modo ATMVP no se puede usar para interpredicir el bloque de imagen comprendido en la imagen de vídeo.

10 El elemento de sintaxis `inter_affine_flag[x0][y0]` se puede usar para especificar si un modo AMVP basado en modelo de movimiento afín se usa para un bloque actual cuando un sector en el que se ubica el bloque actual es un tipo de sector P o un tipo de sector B. Por ejemplo, cuando `inter_affine_flag[x0][y0] = 0`, indica que el modo AMVP basado en modelo de movimiento afín se usa para el bloque actual; o cuando `inter_affine_flag[x0][y0] = 1`, indica que el modo AMVP basado en modelo de movimiento afín no se usa para el bloque actual, pero se puede usar un modo AMVP basado en modelo de movimiento traslacional.

15 El elemento de sintaxis `cu_affine_type_flag[x0][y0]` se puede usar para especificar si un modelo de movimiento afín de 6 parámetros se usa para realizar compensación de movimiento para un bloque actual cuando un sector en el que se ubica el bloque actual es un sector de tipo P o un sector de tipo B. Cuando `cu_affine_type_flag[x0][y0] = 0`, indica que el modelo de movimiento afín de 6 parámetros no se usa para realizar compensación de movimiento para el bloque actual, pero únicamente un modelo de movimiento afín de 4 parámetros se puede usar para realizar compensación de movimiento; o cuando `cu_affine_type_flag[x0][y0] = 1`, indica que el modelo de movimiento afín de 6 parámetros se usa para realizar compensación de movimiento para el bloque actual.

20 Como se muestra En la Tabla 2, cuando `MotionModelIdc[x0][y0] = 1`, indica que se usa un modelo de movimiento afín de 4 parámetros; cuando `MotionModelIdc[x0][y0] = 2`, indica que se usa un modelo de movimiento afín de 6 parámetros; o cuando `MotionModelIdc[x0][y0] = 0`, indica que se usa un modelo de movimiento traslacional.

Tabla 2

30

MotionModelIdc [x0][y0]	Modelo de movimiento para compensación de movimiento (motion model para motion compensation)
0	Traslacional movimiento (translational motion)
1	Movimiento afín de 4 parámetros (4-parameter affine motion)
2	Movimiento afín de 6 parámetros (6-parameter affine motion)

35 La variable `MaxNumMergeCand` se usa para especificar una longitud máxima de una lista de vectores de movimiento candidatos de fusión, la variable `MaxNumSubblockMergeCand` se usa para especificar una longitud máxima de una lista de vectores de movimiento candidatos de fusión basada en subbloques, `inter_pred_idc[x0][y0]` se usa para especificar una dirección de predicción, `PRED_L0` especifica predicción hacia delante, `num_ref_idx_l0_active_minus1` especifica el número de fotogramas de referencia en una lista de fotogramas de referencia hacia delante, `ref_idx_l0[x0][y0]` especifica un valor de índice para un fotograma de referencia hacia delante de un bloque actual, `mvd_coding(x0, y0, 0, 0)` especifica la primera diferencia de vector de movimiento, `mvp_l0_flag[x0][y0]` especifica un valor de índice para una lista de candidatos MVP hacia delante, `PRED_L1` se usa para indicar predicción hacia atrás, `num_ref_idx_l1_active_minus1` indica el número de fotogramas de referencia en una lista de fotogramas de referencia hacia atrás, `ref_idx_l1[x0][y0]` especifica un valor de índice para un fotograma de referencia hacia atrás del bloque actual, y `mvp_l1_flag[x0][y0]` especifica un valor de índice para una lista de candidatos MVP hacia atrás.

40 En la Tabla 1, `ae(v)` representa un elemento de sintaxis codifica usando un codificado aritmético binario adaptativo basado en contexto (CABAC).

A continuación se describe un proceso de interpredicción en detalle, como se muestra en la FIGURA 8A.

50 Etapa 801: Analizar sintácticamente un flujo de bits en función de una estructura de sintaxis mostrada en la Tabla 1, y determinar un modo de interpredicción para un bloque actual.

Si se determina que el modo de interpredicción para el bloque actual es un modo AMVP basado en modelo de movimiento afín, se realiza la etapa 802a.

55

Los elementos de sintaxis `sps_affine_enabled_flag = 1`, `merge_flag = 0`, y `inter_affine_flag = 1` indican que el modo de interpredicción para el bloque actual es el modo AMVP basado en modelo de movimiento afín.

Si se determina que el modo de interpredicción para el bloque actual es un modo de fusión (merge) de subbloques, se realiza la etapa 802b.

5 El elemento de sintaxis `sps_affine_enabled flag = 1` o el elemento de sintaxis `synsps_sbtmvp_enabled flag = 1`, y los elementos de sintaxis `merge_flag = 1` y `merge_subblock flag = 1` indican que el modo de interpredicción para el bloque actual es el modo de fusión de subbloques.

Etapa 802a: Construcción una lista de vectores de movimiento candidatos correspondiente al modo AMVP basado en modelo de movimiento afín. Realizar la etapa 803a.

10 Un vector de movimiento candidato de punto de control del bloque actual se deriva usando un método de predicción de vector de movimiento de punto de control heredado y/o un método de predicción de vector de movimiento de punto de control construido, y se añade a la lista de vectores de movimiento candidatos.

15 La lista de vectores de movimiento candidatos puede comprender una lista de 2-tupla (se usa un modelo de movimiento afín de 4 parámetros para el bloque de codificado actual) o una lista de 3-tupla. La lista de 2-tupla comprende una o más 2-tuplas usadas para construir un modelo de movimiento afín de 4 parámetros. La lista de 3-tupla comprende una o más 3-tuplas usadas para construir un modelo de movimiento afín de 6 parámetros.

20 Opcionalmente, la lista de 2-tupla/3-tupla de vector de movimiento candidato se poda y ordena según una regla particular, y puede truncarse o rellenarse a una cantidad particular.

A1: Se describe un proceso para construir la lista de vectores de movimiento candidatos usando el método de predicción de vector de movimiento de punto de control heredado.

25 La FIGURA 4 se usa como ejemplo. En el ejemplo, en la FIGURA 4, bloques en ubicaciones vecinas alrededor del bloque actual se atraviesan en un orden de A1->B1->B0->A0->B2, para encontrar un bloque codificado afín que incluye un bloque en una ubicación vecina del bloque actual, y para obtener información de movimiento de punto de control del bloque codificado afín. Además, la información de movimiento de punto de control del bloque codificado afín se usa para construir un modelo de movimiento, para derivar información de movimiento de punto de control candidato del bloque actual. Para detalles, consúltense las descripciones anteriores en "3. Método de predicción de vector de movimiento de punto de control heredado". En esta memoria no se describen de nuevo detalles.

35 Por ejemplo, un modelo de movimiento afín usado para el bloque actual es un modelo de movimiento afín de 4 parámetros (esto es, `MotionModelldc = 1`). Si se usa un modelo de movimiento afín de 4 parámetros para un bloque de decodificación afín vecino, se obtienen vectores de movimiento de dos puntos de control del bloque de decodificación afín: un vector de movimiento (v_{x4}, v_{y4}) del punto de control superior izquierdo (x_1, y_4) y un vector de movimiento (v_{x5}, v_{y5}) del punto de control superior derecho (x_5, y_5). El bloque de decodificación afín es un bloque codificado afín para el que se realiza predicción usando un modelo de movimiento afín durante codificación.

40 Vectores de movimiento de dos puntos de control, es decir, los puntos de control superior izquierdo y superior derecho del bloque actual se derivan respectivamente según las fórmulas (6) y (7) correspondientes al modelo de movimiento afín de 4 parámetros, usando el modelo de movimiento afín de 4 parámetros que incluye los dos puntos de control del bloque de decodificación afín vecino.

45 Si se usa un modelo de movimiento afín de 6 parámetros para el bloque de decodificación afín vecino, se obtienen vectores de movimiento de tres puntos de control del bloque de decodificación afín vecino, por ejemplo, un vector de movimiento (v_{x4}, v_{y4}) del punto de control superior izquierdo (x_1, y_4), un vector de movimiento (v_{x5}, v_{y5}) del punto de control superior derecho (x_5, y_5), y un vector de movimiento (v_{x6}, v_{y6}) del vértice inferior izquierdo (x_6, y_6) en la FIGURA 4.

50 Vectores de movimiento de dos puntos de control, es decir, los puntos de control superior izquierdo y superior derecho del bloque actual se derivan respectivamente según las fórmulas (8) y (9) correspondiente al modelo de movimiento afín de 6 parámetros, usando el modelo de movimiento afín de 6 parámetros que incluye los tres puntos de control del bloque de decodificación afín vecino.

55 Por ejemplo, un modelo de movimiento afín usado para un bloque de decodificación actual es un modelo de movimiento afín de 6 parámetros (esto es, `MotionModelldc = 2`). Si un modelo de movimiento afín usado para un bloque de decodificación afín vecino es un modelo de movimiento afín de 6 parámetros, se obtienen vectores de movimiento de tres puntos de control del bloque de decodificación afín vecino, por ejemplo, un vector de movimiento (v_{x4}, v_{y4}) del punto de control superior izquierdo (x_1, y_4), un vector de movimiento (v_{x5}, v_{y5}) del punto de control superior derecho (x_5, y_5), y un vector de movimiento (v_{x6}, v_{y6}) del vértice inferior izquierdo (x_6, y_6) en la FIGURA 4.

65 Vectores de movimiento de tres puntos de control, es decir, los puntos de control superior izquierdo, superior derecho e inferior izquierdo del bloque actual se derivan respectivamente según las fórmulas (8), (9) y (10) correspondiente al

modelo de movimiento afín de 6 parámetros, usando el modelo de movimiento afín de 6 parámetros que incluye los tres puntos de control del bloque de decodificación afín vecino.

5 Si un modelo de movimiento afín usado para el bloque de decodificación afín vecino es un modelo de movimiento afín de 4 parámetros, se obtienen vectores de movimiento de dos puntos de control del bloque de decodificación afín vecino: un vector de movimiento (vx_4, vy_4) del punto de control superior izquierdo (x_1, y_4) y un vector de movimiento (vx_5, vy_5) del punto de control superior derecho (x_5, y_5) .

10 Vectores de movimiento de tres puntos de control, es decir, los puntos de control superior izquierdo, superior derecho e inferior izquierdo del bloque actual se derivan respectivamente según las fórmulas (6) y (7) correspondiente al modelo de movimiento afín de 4 parámetros, usando el modelo de movimiento afín de 4 parámetros que incluye los dos puntos de control del bloque de decodificación afín vecino.

15 Se debe observar que otros modelos de movimiento, ubicaciones de candidatos, y órdenes de búsqueda también se pueden usar en esta solicitud. En esta memoria no se describen detalles. Se debe observar que métodos para representar modelos de movimiento de bloques de codificado vecinos y actuales en función de otros puntos de control también se pueden usar en esta solicitud. En esta memoria no se describen detalles.

20 A2: Se describe un proceso para construir la lista de vectores de movimiento candidatos usando el método de predicción de vector de movimiento de punto de control construido.

25 Por ejemplo, un modelo de movimiento afín usado para un bloque de decodificación actual es un modelo de movimiento afín de 4 parámetros (esto es, $MotionModelldc = 1$), y vectores de movimiento del vértice superior izquierdo y el vértice superior derecho del bloque de codificado actual se determinan en función de información de movimiento de un bloque vecino codificado del bloque de codificado actual. Específicamente, la lista de vectores de movimiento candidatos se pueden construir usando el método de predicción de vector de movimiento de punto de control construido 1 o el método de predicción de vector de movimiento de punto de control construido 2. Para el método específico, consúltense las descripciones anteriores en "4. Método de predicción de vector de movimiento de punto de control construido 1" y "5. Método de predicción de vector de movimiento de punto de control construido 2". En esta memoria no se describen de nuevo detalles.

35 Por ejemplo, un modelo de movimiento afín usado para un bloque de decodificación actual es un modelo de movimiento afín de 6 parámetros (esto es, $MotionModelldc = 2$), y vectores de movimiento del vértice superior izquierdo, el vértice superior derecho y el vértice inferior izquierdo del bloque de codificado actual se determinan en función de información de movimiento de un bloque vecino codificado del bloque de codificado actual. Específicamente, la lista de vectores de movimiento candidatos se pueden construir usando el método de predicción de vector de movimiento de punto de control construido 1 o el método de predicción de vector de movimiento de punto de control construido 2. Para el método específico, consúltense las descripciones anteriores en "4. Método de predicción de vector de movimiento de punto de control construido 1" y "5. Método de predicción de vector de movimiento de punto de control construido 2". En esta memoria no se describen de nuevo detalles.

40 Se debe observar que otras combinaciones de información de movimiento de punto de control también se puede usar en esta solicitud. En esta memoria no se describen detalles.

45 Etapa 803a: Analizar sintácticamente el flujo de bits y determinar un predictor óptimo de vector de movimiento de punto de control. Realizar la etapa 804a.

50 B1: Si el modelo de movimiento afín usado para el bloque de decodificación actual es el modelo de movimiento afín de 4 parámetros ($MotionModelldc = 1$), un número de índice se analiza sintácticamente, y un vector de movimiento predictor óptimo para los dos puntos de control se determinan de la lista de vectores de movimiento candidatos en función del número de índice.

Por ejemplo, el número de índice es `mvp_l0_flag` o `mvp_l1_flag`.

55 B2: Si el modelo de movimiento afín usado para el bloque de decodificación actual es el modelo de movimiento afín de 6 parámetros ($MotionModelldc = 2$), un número de índice se analiza sintácticamente, y un vector de movimiento predictor óptimo para los tres puntos de control se determinan de la lista de vectores de movimiento candidatos en función del número de índice.

60 Etapa 804a: Analizar sintácticamente el flujo de bits y determinar un vector de movimiento de punto de control.

65 C1: Si el modelo de movimiento afín usado para el bloque de decodificación actual es el modelo de movimiento afín de 4 parámetros ($MotionModelldc = 1$), se obtienen diferencias de vector de movimiento de los dos puntos de control del bloque actual al decodificar el flujo de bits, y entonces se obtienen vectores de movimiento de los dos puntos de control en función de las diferencias de vector de movimiento y los predictores de vector de movimiento de los puntos

de control. Usando predicción hacia delante como ejemplo, las diferencias de vector de movimiento de los dos puntos de control son $\text{mvd_coding}(x_0, y_0, 0, 0)$ y $\text{mvd_coding}(x_0, y_0, 0, 1)$, respectivamente.

5 Por ejemplo, diferencias de vector de movimiento del punto de control superior izquierdo y el punto de control superior derecho se obtienen al decodifican el flujo de bits, y se añaden a respectivos predictores de vector de movimiento, para obtener vectores de movimiento del punto de control superior izquierdo y el punto de control superior derecho del bloque actual.

10 C2: Si el modelo de movimiento afín usado para el bloque de decodificación actual es el modelo de movimiento afín de 6 parámetros ($\text{MotionModelIdc} = 2$), se obtienen diferencias de vector de movimiento de los tres puntos de control del bloque actual al decodificar el flujo de bits, y entonces se obtienen vectores de movimiento de los tres puntos de control en función de las diferencias de vector de movimiento y los predictores de vector de movimiento de los puntos de control. Usando predicción hacia delante como ejemplo, diferencias de vector de movimiento de tres puntos de control son $\text{mvd_coding}(x_0, y_0, 0, 0)$, $\text{mvd_coding}(x_0, y_0, 0, 1)$, y $\text{mvd_coding}(x_0, y_0, 0, 2)$, respectivamente.

15 Por ejemplo, diferencias de vector de movimiento del punto de control superior izquierdo, el punto de control superior derecho y el punto de control inferior izquierdo se obtienen al decodificar el flujo de bits, y se añaden a respectivos predictores de vector de movimiento, para obtener vectores de movimiento del punto de control superior izquierdo, el punto de control superior derecho y el punto de control inferior izquierdo del bloque actual.

20 Etapa 805a: Obtener un vector de movimiento de cada subbloque del bloque actual en función de información de movimiento de punto de control y el modelo de movimiento afín usado para el bloque de decodificación actual.

25 Un subbloque en el actual bloque de decodificación afín puede ser equivalente a una unidad de compensación de movimiento, y la anchura y la altura del subbloque son menores que la anchura y la altura del bloque actual. Información de movimiento de una muestra en una ubicación preestablecida en una unidad de compensación de movimiento se puede usar para representar información de movimiento de todas las muestras de la unidad de compensación de movimiento. Asumiendo que el tamaño de la unidad de compensación de movimiento es $M \times N$, la muestra en la ubicación preestablecida puede ser una muestra central ($M/2, N/2$), la muestra superior izquierda ($0, 0$), la muestra superior derecha ($M-1, 0$), o una muestra en otra ubicación en la unidad de compensación de movimiento. La muestra central de la unidad de compensación de movimiento se usa como ejemplo para la descripción a continuación, como se muestra en la FIGURA 8C. En la FIGURA 8C, V_0 representa un vector de movimiento del punto de control superior izquierdo, y V_1 representa un vector de movimiento del punto de control superior derecho. Cada caja pequeña representa una unidad de compensación de movimiento.

35 Coordenadas de una muestra central de una unidad de compensación de movimiento respecto a la muestra superior izquierda del actual bloque de decodificación afín se calculan según la Fórmula (31). En la Fórmula (31), i representa la i -ésima unidad de compensación de movimiento en la dirección horizontal (de izquierda a derecha), j representa la j -ésima unidad de compensación de movimiento en la dirección vertical (de parte superior a parte inferior), y $(x_{(i,j)}, y_{(i,j)})$ representa coordenadas de la muestra central de la (i, j) -ésima unidad de compensación de movimiento respecto al punto de control superior izquierdo muestra del actual bloque de decodificación afín.

45 Si el modelo de movimiento afín usado para el actual bloque de decodificación afín es el modelo de movimiento afín de 6 parámetros, $(x_{(i,j)}, y_{(i,j)})$ se sustituye en la Fórmula (32) correspondiente al modelo de movimiento afín de 6 parámetros para obtener un vector de movimiento $(vx_{(i,j)}, vy_{(i,j)})$ de una muestra central de cada unidad de compensación de movimiento, y el vector de movimiento se usa como vectores de movimiento de todas las muestras de la unidad de compensación de movimiento.

50 Si el modelo de movimiento afín usado para el actual bloque de decodificación afín es el modelo de movimiento afín de 4 parámetros, $(x_{(i,j)}, y_{(i,j)})$ se sustituye en la Fórmula (33) correspondiente al modelo de movimiento afín de 4 parámetros para obtener un vector de movimiento $(vx_{(i,j)}, vy_{(i,j)})$ de una muestra central de cada unidad de compensación de movimiento, y el vector de movimiento se usa como vectores de movimiento de todas las muestras de la unidad de compensación de movimiento.

$$\begin{cases} x_{(i,j)} = M \times i + \frac{M}{2}, i = 0, 1, \dots \\ y_{(i,j)} = N \times j + \frac{N}{2}, j = 0, 1, \dots \end{cases} \quad (31)$$

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W} x + \frac{vx_2 - vy_0}{H} y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W} x + \frac{vy_2 - vx_0}{H} y + vy_0 \end{cases} \quad (32)$$

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W} x - \frac{vy_1 - vy_0}{W} y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W} x + \frac{vx_1 - vx_0}{W} y + vy_0 \end{cases} \quad (33)$$

Etapa 806a: Realizar compensación de movimiento para cada subbloque en función de la determinado vector de movimiento del subbloque, para obtener un valor de predicción de muestra del subbloque.

5 Etapa 802b: Construir una lista de candidatos de información de movimiento correspondiente a un modo de fusión (merge) de subbloques.

10 Específicamente, la lista de candidatos de información de movimiento correspondiente al modo de fusión de subbloque (sub-block based merging candidate list) se pueden construir usando una o más de predicción de vector de movimiento temporal avanzada, un método de predicción de vector de movimiento de punto de control heredado, un método de predicción de vector de movimiento de punto de control construido, o un método plano. En esta realización de esta solicitud, la lista de candidatos de información de movimiento correspondiente al modo de fusión de subbloque puede denominarse lista de candidatos de fusión basada en subbloques por abreviar.

15 Opcionalmente, la lista de candidatos de información de movimiento se poda y ordena según una regla particular, y puede truncarse o rellenarse a una cantidad particular.

20 D0: Si `sps_sbtmvp_enabled` flag = 1, un vector de movimiento candidato se obtiene usando la ATMVP, y se añade a la lista de candidatos de fusión basada en subbloques. Para detalles, consúltense las descripciones anteriores en "6. Predicción de vector de movimiento temporal avanzada".

25 D1: Si `sps_affine_enabled` flag = 1, se deriva información de movimiento de punto de control candidato del bloque actual usando el método de predicción de vector de movimiento de punto de control heredado, y se añade a la lista de candidatos de fusión basada en subbloques. Para detalles, consúltense las descripciones anteriores en "3. Método de predicción de vector de movimiento de punto de control heredado".

30 Por ejemplo, en la FIGURA 6C, bloques en ubicaciones vecinas alrededor del bloque actual se atraviesan en un orden de A1->B1->B0->A0->B2, para encontrar un bloque codificado afín que incluye un bloque en una ubicación vecina del bloque actual, y para obtener información de movimiento de punto de control del bloque codificado afín. Además, un modelo de movimiento correspondiente al bloque codificado afín se usa para derivar información de movimiento de punto de control candidato del bloque actual.

35 Si la lista de candidatos de fusión basada en subbloques está vacía, la información de movimiento de punto de control candidato se añade a la lista de candidatos de fusión basada en subbloques. De otro modo, información de movimiento en la lista de candidatos de fusión basada en subbloques se atraviesa secuencialmente para comprobar si información de movimiento que es la misma que la información de movimiento de punto de control candidato existe en la lista de candidatos de fusión basada en subbloques. Si en la lista de candidatos de fusión basada en subbloques no existe información de movimiento que es la misma que la información de movimiento de punto de control candidato, la información de movimiento de punto de control candidato se añade a la lista de candidatos de fusión basada en subbloques.

45 Determinar si dos elementos de información de movimiento candidato son iguales se va a implementar determinando si fotografías de referencia hacia delante y hacia atrás de los dos elementos de información de movimiento candidato y componentes horizontal y vertical de cada vector de movimiento hacia delante y vector de movimiento hacia atrás son iguales. Los dos elementos de información de movimiento candidato se consideran diferentes únicamente cuando todos estos elementos son diferentes.

50 Si una cantidad de elementos de información de movimiento en la lista de candidatos de fusión basada en subbloques alcanza una longitud máxima de lista `MaxNumSubblockMergeCand` (`MaxNumSubblockMergeCand` es un entero positivo tal como 1, 2, 3, 4 o 5, donde 5 se usa como ejemplo en las siguientes descripciones, y en esta memoria no se en esta memoria no se describen detalles), se completar la construcción de la lista de candidatos. De otro modo, se atraviesa un bloque en una siguiente ubicación vecina.

55 D2: Si `sps_affine_enabled` flag = 1, información de movimiento de punto de control candidato del bloque actual se deriva usando el método de predicción de vector de movimiento de punto de control construido, y se añade a la lista de candidatos de fusión basada en subbloques, como se muestra en la FIGURA 8B.

60 Etapa 801c: Obtener información de movimiento de puntos de control del bloque actual. Por ejemplo, para detalles, consúltense la etapa 601 en las descripciones anteriores en "5. Método de predicción de vector de movimiento de punto de control construido 2". En esta memoria no se describen de nuevo los detalles.

65 Etapa 802c: Combinar la información de movimiento de los puntos de control para obtener información de movimiento de punto de control construido. Para detalles, consúltense la etapa 602 en la FIGURA 6D. En esta memoria no se describen de nuevo los detalles.

Etapa 803c: Añadir la información de movimiento de punto de control construido a la lista de candidatos de fusión basada en subbloques.

5 Si la longitud de la lista de candidatos de fusión basada en subbloques es menor que una longitud máxima de lista MaxNumSubblockMergeCand, las combinaciones de la información de movimiento de los puntos de control se atraviesan en un orden preestablecido, y se usa una combinación válida resultante como la información de movimiento de punto de control candidato. En este caso, si la lista de candidatos de fusión basada en subbloques está vacía, la información de movimiento de punto de control candidato se añade a la lista de candidatos de fusión basada en subbloques. De otro modo, información de movimiento en la lista de vectores de movimiento candidatos se atraviesa
10 secuencialmente para comprobar si información de movimiento que es la misma que la información de movimiento de punto de control candidato existe en la lista de candidatos de fusión basada en subbloques. Si en la lista de candidatos de fusión basada en subbloques no existe información de movimiento que es la misma que la información de movimiento de punto de control candidato, la información de movimiento de punto de control candidato se añade a la lista de candidatos de fusión basada en subbloques.

15 Por ejemplo, un orden preestablecido es de la siguiente manera: Afín (CP1, CP2, CP3) -> Afín (CP1, CP2, CP4) -> Afín (CP1, CP3, CP4) -> Afín (CP2, CP3, CP4) -> Afín (CP1, CP2) -> Afín (CP1, CP3). Hay seis combinaciones en total.

20 Por ejemplo, si `sps_affine_type_flag = 1`, un orden preestablecido es de la siguiente manera: Afín (CP1, CP2, CP3) -> Afín (CP1, CP2, CP4) -> Afín (CP1, CP3, CP4) -> Afín (CP2, CP3, CP4) -> Afín (CP1, CP2) -> Afín (CP1, CP3). Hay seis combinaciones en total. Un orden para añadir las seis combinaciones a la lista de vectores de movimiento candidatos no se limita específicamente en esta realización de esta solicitud.

25 Si `sps_affine_type_flag = 0`, un orden preestablecido es de la siguiente manera: Afín (CP1, CP2) -> Afín (CP1, CP3). Hay dos combinaciones en total. Un orden para añadir los dos combinaciones a la lista de vectores de movimiento candidatos no se limita específicamente en esta realización de esta solicitud.

30 Si no hay disponible información de movimiento de punto de control correspondiente a una combinación, la combinación se considera no disponible. Si hay disponible una combinación, se determina un índice de fotograma de referencia de la combinación (cuando hay dos puntos de control, un índice de fotograma de referencia mínimo se selecciona como índice de fotograma de referencia de la combinación; cuando hay más de dos puntos de control, un índice de fotograma de referencia con una frecuencia de presencia máxima se selecciona como el índice de fotograma de referencia de la combinación; y si una pluralidad de índices de fotogramas de referencia tienen una misma frecuencia de presencia, un índice de fotograma de referencia mínimo se selecciona como índice de fotograma de referencia de la combinación). Además se escalan vectores de movimiento de punto de control. Si información de movimiento de todas los puntos de control tras escalado es la misma, la combinación es inválida.

35 D3: Opcionalmente, si `sps_planar_enabled_flag = 1`, se añade información de movimiento construido usando la ATMVP a la lista de candidatos de fusión basada en subbloques. Para detalles, consúltense las descripciones anteriores en "7. Predicción de vector de movimiento plano".

40 Opcionalmente, en esta realización de esta solicitud, la lista de vectores de movimiento candidatos puede rellenarse. Por ejemplo, tras el anterior proceso de recorrido a través, si la longitud de la lista de vectores de movimiento candidatos es menor que la longitud máxima de lista MaxNumSubblockMergeCand, la lista de vectores de movimiento candidatos puede rellenarse hasta que la longitud de lista es igual a MaxNumSubblockMergeCand.

45 El relleno puede realizarse usando un método de relleno de vector de movimiento cero, o usando un método para combinar o promediar ponderando información existente de movimiento candidato en la lista existente. Se debe observar que otros métodos para relleno la lista de vectores de movimiento candidatos también se pueden usar en esta solicitud. En esta memoria no se describen detalles.

Etapa 803b: Analizar sintácticamente el flujo de bits y determinar información de movimiento de punto de control óptimo.

55 Un número de índice `merge_subblock_idx` se analiza sintácticamente, y la información de movimiento óptimo se determina de la lista de candidatos de fusión basada en subbloques en función del número de índice.

60 `merge_subblock_idx` usualmente se binariza usando un código TR (unario truncado, código unario truncado). En otras palabras, `merge_subblock_idx` se mapea a diferentes cadenas binarias en función de un valor máximo de índice. El valor máximo de índice se preconfigura o se transmite. Por ejemplo, si el valor máximo de índice es 4, `merge_subblock_idx` se binariza según la Tabla 3.

Tabla 3

Índice	Cadena binaria				
0	0				
1	1	0			
2	1	1	0		
3	1	1	1	0	
4	1	1	1	1	

5 Cuando el valor máximo de índice es 2, merge_subblock_idx se binariza según la Tabla 4.

Tabla 4

Índice	Cadena binaria				
0	0				
1	1	0			
2	1	1			

10 Si merge_subblock_idx se transmite usando una cadena binaria, un lado de decodificador puede determinar el número de índice en función del valor máximo de índice y Tabla 2 o Tabla 3. Por ejemplo, el valor máximo de índice es 4. Cuando se realiza decodificación para obtener el número de índice, si se obtiene 0 o el número de índice obtenido a través de decodificación es igual al valor máximo de índice, se determina el número de índice. Por ejemplo, cuando se realiza decodificación para obtener el número de índice, si el primer bit es 0, la decodificación que se realiza para obtener el número de índice finaliza, y el número de índice se determina que es 0. Para otro ejemplo, si el primer bit es 1 y el segundo bit es 0, la decodificación que se realiza para obtener el número de índice finaliza, y el número de índice se determina que es 1.

15 El establecimiento del valor máximo de índice y una tabla de binarización no se limitan específicamente en esta realización de esta solicitud.

20 Etapa 804b: Si la información de movimiento óptimo es ATMVP o información de movimiento plano, obtener un vector de movimiento de cada subbloque usando directamente la ATMVP o método plano.

25 Si un modo de movimiento indicado por la información de movimiento óptimo es un modo afín, el vector de movimiento de cada subbloque del bloque actual se obtiene en función de la información de movimiento de punto de control óptimo y un modelo de movimiento afín usado para un bloque de decodificación actual. Esto proceso es el mismo que la etapa 805a.

30 Etapa 805b: Realizar compensación de movimiento para cada subbloque en función de la determinado vector de movimiento del subbloque, para obtener un valor de predicción de muestra del subbloque.

35 En la tecnología existente, no hay manera factible de determinar una longitud máxima (MaxNumSubblockMergeCand) de la lista de vectores de movimiento candidatos correspondiente al modo de fusión de subbloque.

40 En vista de esto, las realizaciones de esta solicitud proporcionan un método y un aparato de predicción de imágenes de vídeo, para proporcionar una manera de determinar una longitud máxima (MaxNumSubblockMergeCand) de una lista de vectores de movimiento candidatos correspondiente a un modo de fusión de subbloque. El método y el aparato se basan en un mismo concepto inventivo. Como un principio de resolución de problemas del método es similar al del aparato, puede hacerse referencia mutuo entre implementaciones del aparato y el método. No se proporcionan descripciones repetidas.

45 La manera de determinar la longitud máxima de la lista de vectores de movimiento candidatos correspondiente al modo de fusión de subbloque se describe a continuación en detalle en dos casos.

En un primer caso, el modo de fusión de subbloque puede comprender al menos uno de un modo afín, un modo de predicción de vector de movimiento temporal avanzada o un modo de predicción de vector de movimiento plano.

En un segundo caso, un modo de predicción de vector de movimiento plano no se considera que esté presente en el modo de fusión de subbloque. En otras palabras, el modo de fusión de subbloque puede comprender al menos uno de un modo afín o un modo de predicción de vector de movimiento temporal avanzada.

5 A continuación se describen implementaciones de esta solicitud en detalle desde una perspectiva de un lado de decodificador con referencia a los dibujos adjuntos. Específicamente, la manera puede ser realizada por un decodificador de vídeo 30, o puede ser implementada por un módulo de compensación de movimiento en un decodificador de vídeo, o puede ser realizada por un procesador.

10 Para el primer caso, varias posibles implementaciones, una primera implementación a una quinta implementación, se proporcionan como ejemplos.

La FIGURA 9 muestra descripciones de la primera implementación.

15 Etapa 901: Analizar sintácticamente un primer indicador de un flujo de bits. Realizar S902 o 5904.

El primer indicador se usa para indicar si un modo candidato usado para interpredecir un bloque a procesar comprende el modo afín. En otras palabras, el indicador 1 se usa para indicar si el modo afín puede (o se permite) ser usado para realizar compensación de movimiento en un bloque a procesar.

20 Por ejemplo, el primer indicador se puede configurar en un SPS en el flujo de bits. Sobre esta base, analizar sintácticamente el primer indicador del flujo de bits puede implementarse del siguiente manera: analizar sintácticamente el indicador 1 del SPS en el flujo de bits. Como alternativa, el primer indicador se puede configurar en una cabecera de sector de un sector en el flujo de bits, el bloque a procesar se comprende en el sector. En función de esto, analizar sintácticamente el primer indicador del flujo de bits puede implementarse en el siguiente manera: analizar sintácticamente el primer indicador de la cabecera de sector del sector en el flujo de bits, el bloque a procesar se comprende en el sector.

30 Por ejemplo, el primer indicador puede ser representado por un elemento de sintaxis `sps_affine_enabled_flag`, y `sps_affine_enabled_flag` se usa para especificar si un modo afín se puede usar para interpredecir un bloque de imagen comprendido en una imagen de vídeo. Por ejemplo, cuando `sps_affine_enabled_flag = 0`, indica que el modo afín no se puede usar para interpredecir el bloque de imagen comprendido en la imagen de vídeo. Cuando `sps_affine_enabled_flag = 1`, indica que un modelo de movimiento afín se puede usar para interpredecir el bloque de imagen comprendido en la imagen de vídeo.

35 S902: Cuando el primer indicador indica que el modo candidato usado para interpredecir el bloque a procesar comprende el modo afín, analizar sintácticamente un segundo indicador del flujo de bits, donde el segundo indicador se usa para indicar (o determinar) una longitud máxima de una primera lista de vectores de movimiento candidatos, y la primera lista de vectores de movimiento candidatos es una lista de vectores de movimiento candidatos construido para el bloque a procesar usando un modo de predicción de fusión de subbloque. La primera lista de vectores de movimiento candidatos se denomina `MaxNumSubblockMergeCand`.

40 Por ejemplo, el segundo indicador se puede configurar en el SPS, un PPS, o la cabecera de sector. En función de esto, analizar sintácticamente el segundo indicador del flujo de bits puede implementarse en el siguiente manera: analizar sintácticamente el segundo indicador del conjunto de parámetros de secuencia en el flujo de bits; o analizar sintácticamente el segundo indicador de la cabecera de sector del sector, que incluye el bloque a procesar, en el flujo de bits.

45 Por ejemplo, el segundo indicador puede representarse por `K_minus_max_num_subblock_merge_cand`.

50 Por ejemplo, se permite que un valor de `K_minus_max_num_subblock_merge_cand` esté en un intervalo de 0 a 4.

Por ejemplo, se permite que un valor máximo de `MaxNumSubblockMergeCand` sea 5.

55 Cuando se permite que el valor máximo de `MaxNumSubblockMergeCand` sea 5, el segundo indicador puede representarse por `five_minus_max_num_subblock_merge_cand`.

La Tabla 5 muestra un ejemplo de una estructura de sintaxis para analizar sintácticamente el segundo indicador.

60

Tabla 5

slice_header() {	Descriptor
...	
if(sps_affine_enable_flag)	
five_minus_max_num_subblock_merge_cand	ue(v)
...	
slice_header() {	Descriptor
}	

5 S903: Determinar la longitud máxima de la primera lista de vectores de movimiento candidatos en función del segundo indicador.

En un ejemplo, cuando el primer indicador indica que el modo candidato usado para interpretar el bloque a procesar comprende el modo afin, la longitud máxima (MaxNumSubblockMergeCand) de la primera lista de vectores de movimiento candidatos puede obtenerse según la siguiente fórmula: $MaxNumSubblockMergeCand = K - K_minus_max_num_subblock_merge_cand$,

15 donde MaxNumSubblockMergeCand representa la longitud máxima de la primera lista de vectores de movimiento candidatos, K_minus_max_num_subblock_merge_cand representa el segundo indicador, y K es un entero no negativo preestablecido.

20 S904: Cuando el primer indicador indica que el modo candidato usado para interpretar el bloque a procesar únicamente comprende el modo de predicción de vector de movimiento traslacional, y el tercer indicador se usa para indicar que el modo ATMVP está presente en el modo de predicción de fusión de subbloque, determinar un tercer número en función del tercer indicador, y determinar la longitud máxima de la primera lista de vectores de movimiento candidatos en función del tercer número.

25 Que el modo candidato usado para interpretar el bloque a procesar únicamente comprende el modo de predicción de vector de movimiento traslacional significa que el modo candidato usado para interpretar el bloque a procesar no puede (no se permite) comprender el modo afin. El tercer indicador se usa para indicar si el modo ATMVP está presente en el modo de predicción de fusión de subbloque. En otras palabras, el tercer indicador se usa para indicar si el modo ATMVP se permite para ser usado para interpretar el bloque a procesar. El tercer indicador se puede configurar en el SPS, la PPS, o la cabecera de sector.

30 En un ejemplo, cuando el primer indicador indica que el modo candidato usado para interpretar el bloque a procesar únicamente comprende el modo de predicción de vector de movimiento traslacional, y el tercer indicador se usa para indicar que el modo ATMVP está presente en el modo de predicción de fusión de subbloque, la longitud máxima (MaxNumSubblockMergeCand) de la primera lista de vectores de movimiento candidatos es igual al tercer número.

35 Por ejemplo, el tercer indicador puede ser representado por sps_sbtmvp_enabled_flag. Por ejemplo, cuando sps_sbtmvp_enabled_flag es igual a un primer valor, indica que el modo ATMVP no está presente en el modo de predicción de fusión de subbloque; o cuando sps_sbtmvp_enabled_flag es igual a un segundo valor, indica que el modo ATMVP está presente en el modo de predicción de fusión de subbloque. Por ejemplo, el primer valor es igual a 0, y el segundo valor es igual a 1.

40 Por ejemplo, el tercer número se puede usar para indicar el número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo ATMVP. Por ejemplo, cuando el primer indicador indica que el modo candidato usado para interpretar el bloque a procesar únicamente comprende el modo de predicción de vector de movimiento traslacional, si sps_sbtmvp_enabled_flag = 0, el tercer número es igual a 0, o si sps_sbtmvp_enabled_flag = 1, el tercer número es igual al número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo ATMVP. Adicionalmente, cuando el número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo ATMVP es 1, el tercer número puede ser igual a un valor de sps_sbtmvp_enabled_flag. Por ejemplo, si sps_sbtmvp_enabled_flag = 0, el tercer número es igual a 0; o si sps_sbtmvp_enabled_flag = 1, el tercer número es igual a 1..

50 Que se permite que el valor máximo de MaxNumSubblockMergeCand sea 5 se usa como ejemplo. Si sps_affine_enable_flag = 0, MaxNumSubblockMergeCand se obtiene según la siguiente fórmula:

$$MaxNumSubblockMergeCand = sps_sbtmvp_enabled_flag.$$

Si `sps_affine_enable_flag = 1`, `MaxNumSubblockMergeCand` se obtiene según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_mergc_cand}.$$

Por ejemplo, `five_minus_max_num_subblock_merge_cand` se puede definir como sustraer una longitud máxima de una lista de predicción de vectores de movimiento de fusión basada en subbloques en un sector de 5 (`five_minus_max_num_subblock_merge_cand` especifica el número máximo de candidatos de predicción de vector de movimiento (MVP) de fusión basada en subbloques soportados en el sector sustraído de 5).

Según la presente invención, el número máximo de MVP de fusión basada en subbloques candidatos, `MaxNumSubblockMergeCand`, se deriva de la siguiente manera:

- Si `sps_affine_enabled_flag` es igual a 0:

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag};$$

- De lo contrario (`sps_affine_enabled_flag` es igual a 1):

$$\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_mergc_cand}.$$

Un valor de `MaxNumSubblockMergeCand` estará en el intervalo de 0 a 5, inclusive.

La FIGURA 10 muestra descripciones de la segunda implementación.

S1001: Para detalles de S1001, consúltese S901. En esta memoria no se describen de nuevo los detalles. Realizar S1002 o S1004.

S1002: Cuando el primer indicador indica que el modo candidato usado para interpredcir el bloque a procesar comprende el modo afín, analizar sintácticamente un segundo indicador, y un tercer indicador del flujo de bits, donde el segundo indicador se usa para indicar (o determinar) una longitud máxima de una primera lista de vectores de movimiento candidatos, y la primera lista de vectores de movimiento candidatos es una lista de vectores de movimiento candidatos construido para el bloque a procesar usando un modo de predicción de fusión de subbloque.

S1003: Cuando el tercer indicador indica que el modo de predicción de vector de movimiento temporal avanzada no está presente en el modo de predicción de fusión de subbloque, determinar un primer número en función del tercer indicador, y determinar la longitud máxima de la primera lista de vectores de movimiento candidatos en función del segundo indicador y el primer número.

Para explicaciones del primer indicador, el segundo indicador, y el tercer indicador, consúltese la realización correspondiente a la FIGURA 9. En esta memoria no se describen de nuevo detalles.

Cuando el tercer indicador indica que el modo ATMVP no está presente en el modo de predicción de fusión de subbloque, el primer número puede ser igual al número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo ATMVP. Por ejemplo, el tercer indicador puede ser representado por `sps_sbtmvp_enabled_flag`. Cuando `sps_sbtmvp_enabled_flag = 0`, indica que el modo de predicción de vector de movimiento temporal avanzada no está presente en el modo de predicción de fusión de subbloque. En este caso, el primer número es igual al número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo ATMVP. Por el contrario, cuando `sps_sbtmvp_enabled flag = 1`, indica que el modo de predicción de vector de movimiento temporal avanzada está presente en el modo de predicción de fusión de subbloque. En este caso, el primer número es igual a 0. Por ejemplo, el número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo ATMVP puede ser 1. En este caso, el primer número puede ser igual a un valor del tercer indicador. Para los detalles, consúltese las descripciones en la realización correspondiente a la FIGURA 9. En esta memoria no se describen de nuevo los detalles.

En un ejemplo posible, cuando el primer indicador indica que el modo candidato usado para interpredcir el bloque a procesar comprende el modo afín, la longitud máxima de la primera lista de vectores de movimiento candidatos puede obtenerse según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = K - K_minus_max_num_subblock_merge_cand - L1,$$

donde `MaxNumSubblockMergeCand` representa la longitud máxima de la primera lista de vectores de movimiento candidatos, `K_minus_max_num_subblock_merge_cand` representa el segundo indicador, `L1` representa el primer número, y `K` es un entero no negativo preestablecido.

Por ejemplo, puede permitirse que un valor de `K_minus_max_num_subblock_merge_cand` esté en un intervalo de 0 a 3.

Por ejemplo, se permite que un valor máximo de `MaxNumSubblockMergeCand` sea 5.

5 Cuando se permite que el valor máximo de `MaxNumSubblockMergeCand` sea 5, el segundo indicador puede representarse por `five_minus_max_num_subblock_merge_cand`.

En un ejemplo, `L1` puede obtenerse según la siguiente fórmula:

10
$$L1 = \text{sps_sbtmvp_enabled_flag} == 1 ? 0 : 1. \text{ If } \text{sps_sbtmvp_enabled_flag} = 1, L1 = 0. \text{ If } \text{sps_sbtmvp_enabled_flag} = 0, L1 = 1.$$

S 1004: Para detalles de S 1004, consúltese S904. En esta memoria no se describen de nuevo los detalles.

15 Que se permite que el valor máximo de `MaxNumSubblockMergeCand` sea 5 se usa como ejemplo. Si `sps_affine_enable_flag = 0`, `MaxNumSubblockMergeCand` se obtiene según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag}.$$

20 Si `sps_affine_enable_flag = 1`, `MaxNumSubblockMergeCand` se obtiene según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_merge_cand} - (\text{sps_sbtmvp_enabled_flag} == 1 ? 0 : 1).$$

25 Por ejemplo, `five_minus_max_num_subblock_merge_cand` se puede definir como sustraer una longitud máxima de una lista de predicción de vectores de movimiento de fusión basada en subbloques en un sector de 5 (`five_minus_max_num_subblock_merge_cand` especifica el número máximo de candidatos de predicción de vector de movimiento (MVP) de fusión basada en subbloques soportados en el sector sustraído de 5). El número máximo de candidatos de MVP de fusión basada en subbloques, `MaxNumSubblockMergeCand`, se deriva de la siguiente manera:

30 - Si `sps_affine_enabled_flag` es igual a 0:

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag};$$

35 - De lo contrario (`sps_affine_enabled_flag` es igual a 1):

$$\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_merge_cand} - (\text{sps_sbtmvp_enabled_flag} == 1 ? 0 : 1).$$

Un valor de `MaxNumSubblockMergeCand` estará en el intervalo de 0 a 5, inclusive.

40 La FIGURA 11 muestra descripciones de la tercera implementación.

S1101 a S1103: Para detalles de S1101 a S1103, consúltese de S901 a S903. En esta memoria no se describen de nuevo los detalles.

45 Se debe observar que, en la tercera implementación, puede permitirse que un valor máximo de `MaxNumSubblockMergeCand` sea 5. Por ejemplo, el segundo indicador puede representarse por `K_minus_max_num_subblock_merge_cand`, donde se permite que un valor de `K_minus_max_num_subblock_merge_cand` esté en el intervalo de 0 a 4. Cuando se permite que el valor máximo de `MaxNumSubblockMergeCand` sea 5, el segundo indicador puede representarse por `five_minus_max_num_subblock_merge_cand`.

50 S1104: Cuando el primer indicador indica que el modo candidato usado para interpredcir el bloque a procesar comprende únicamente el modo de predicción de vector de movimiento traslacional, analizar sintácticamente un tercer indicador y un cuarto indicador del flujo de bits. Realizar S1105, S1106 o S1107.

55 El tercer indicador se usa para indicar un estado de presencia del modo ATMVP en el modo de predicción de fusión de subbloque. Para descripciones relacionadas del tercer indicador, consúltense las descripciones en la realización correspondiente a la FIGURA 9. En esta memoria no se describen de nuevo detalles.

El cuarto indicador se usa para indicar un estado de presencia del modo de predicción de vector de movimiento traslacional (PLANO) en el modo de predicción de fusión de subbloque. En otras palabras, el cuarto indicador se usa para indicar si se permite usar el modo plano para interpredecir el bloque a procesar.

5 S1105: Cuando el tercer indicador indica que el modo ATMVP está presente en el modo de predicción de fusión de subbloque, y el cuarto indicador indica que el modo plano no está presente en el modo de predicción de fusión de subbloque, determinar un tercer número en función del tercer indicador, y determinar la longitud máxima de la primera lista de vectores de movimiento candidatos en función de únicamente el tercer número.

10 Por ejemplo, cuando el cuarto indicador es un tercer valor, indica que el modo plano no está presente en el modo de predicción de fusión de subbloque; o cuando el cuarto indicador es un cuarto valor, indica que el modo plano está presente en el modo de predicción de fusión de subbloque. Por ejemplo, el tercer valor es igual a 0, y el cuarto valor es igual a 1.. Por ejemplo, el cuarto indicador se puede configurar en un SPS, un PPS, o una cabecera de sector. El cuarto indicador puede ser representado por `sps_planar_enabled_flag`.

15 S1106: Cuando el tercer indicador indica que el modo ATMVP no está presente en el modo de predicción de fusión de subbloque, y el cuarto indicador indica que el modo plano está presente en el modo de predicción de fusión de subbloque, determinar un cuarto número en función del cuarto indicador, y determinar la longitud máxima de la primera lista de vectores de movimiento candidatos en función del cuarto número.

20 Por ejemplo, cuando el cuarto indicador indica que el modo plano está presente en el modo de predicción de fusión de subbloque, el cuarto número es igual al número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo plano.

25 Por ejemplo, en S1106, la longitud máxima de la primera lista de vectores de movimiento candidatos es igual al cuarto número. Por ejemplo, si el número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo plano es 1, la longitud máxima de la primera lista de vectores de movimiento candidatos es 1. Para otro ejemplo, cuando el cuarto indicador es 1, indica que el modo plano está presente en el modo de predicción de fusión de subbloque. En este caso, la longitud máxima de la primera lista de vectores de movimiento candidatos es igual al cuarto indicador.

30 S 1107: Cuando el tercer indicador indica que el modo ATMVP está presente en el modo de predicción de fusión de subbloque, y el cuarto indicador indica que el modo plano está presente en el modo de predicción de fusión de subbloque, determinar un tercer número en función del tercer indicador, determinar un cuarto número en función del cuarto indicador, y determinar la longitud máxima de la primera lista de vectores de movimiento candidatos en función del tercer número y el cuarto número.

35 En un ejemplo, cuando el primer indicador indica que el modo candidato usado para interpredecir el bloque a procesar únicamente comprende el modo de predicción de vector de movimiento traslacional, el número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo plano es 1, y el número máximo de vectores de movimiento que son soportados en predicción realizada usando en el modo ATMVP es 1, si el tercer indicador es 1, indica que el modo ATMVP está presente en el modo de predicción de fusión de subbloque; o si el tercer indicador es 0, indica que el modo ATMVP no está presente en el modo de predicción de fusión de subbloque; y si el cuarto indicador es 1, indica que el modo plano está presente en el modo de predicción de fusión de subbloque; o si el cuarto indicador es 0, indica que el modo plano está presente en el modo de predicción de fusión de subbloque. En este caso, la longitud máxima de la primera lista de vectores de movimiento candidatos puede ser igual a una suma del tercer indicador y el cuarto indicador.

40 El tercer indicador se representa por `sps_sbtmvp_enabled_flag`, y el cuarto indicador se representa por `sps_planar_enabled_flag`. La longitud máxima de la primera lista de vectores de movimiento candidatos puede obtenerse según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag} + \text{sps_planar_enabled_flag}.$$

45 Cuando `sps_sbtmvp_enabled_flag = 1` (que indica que el modo ATMVP está presente en el modo de predicción de fusión de subbloque), y `sps_planar_enabled_flag = 0` (que indica que el modo plano no está presente en el modo de predicción de fusión de subbloque), `MaxNumSubblockMergeCand = sps_sbtmvp_enabled_flag = 1`, que corresponde a S 1105. Cuando `sps_sbtmvp_enabled_flag = 0` (que indica que el modo ATMVP no está presente en el modo de predicción de fusión de subbloque), y `sps_planar_enabled_flag = 1` (que indica que el modo plano está presente en el modo de predicción de fusión de subbloque), `MaxNumSubblockMergeCand = sps_planar_enabled_flag = 1`, que corresponde a S1106. Cuando `sps_sbtmvp_enabled_flag = 1`, y `sps_planar_enabled_flag = 1`, `MaxNumSubblockMergeCand = sps_sbtmvp_enabled_flag + sps_planar_enabled_flag = 2`, que corresponde a S 1107.

50 Desde luego, cuando el primer indicador indica que el modo candidato usado para interpredecir el bloque a procesar únicamente comprende el modo de predicción de vector de movimiento traslacional, el tercer indicador indica que el

65

modo ATMVP no está presente en el modo de predicción de fusión de subbloque, y el cuarto indicador indica que el modo plano no está presente en el modo de predicción de fusión de subbloque, $\text{MaxNumSubblockMergeCand} = 0$.

Que se permite que el valor máximo de $\text{MaxNumSubblockMergeCand}$ sea 5 se usa como ejemplo.

5

Si $\text{sps_affine_enable_flag} = 0$, $\text{MaxNumSubblockMergeCand}$ se obtiene según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag} + \text{sps_planar_enabled_flag}.$$

10

Si $\text{sps_affine_enable_flag} = 1$, $\text{MaxNumSubblockMergeCand}$ se obtiene según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_merge_cand}.$$

15

Por ejemplo, $\text{five_minus_max_num_subblock_merge_cand}$ se puede definir como sustraer una longitud máxima de una lista de predicción de vectores de movimiento de fusión basada en subbloques en un sector de 5 ($\text{five_minus_max_num_subblock_merge_cand}$ especifica el número máximo de candidatos de predicción de vector de movimiento (MVP) de fusión basada en subbloques soportados en el sector sustraído de 5).

20

El número máximo de candidatos de MVP de fusión basada en subbloques, $\text{MaxNumSubblockMergeCand}$, se deriva de la siguiente manera:

- Si $\text{sps_affine_enabled_flag}$ es igual a 0:

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag} + \text{sps_planar_enabled_flag};$$

25

- De lo contrario ($\text{sps_affine_enabled_flag}$ es igual a 1):

$$\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_merge_cand}.$$

30

Un valor de $\text{MaxNumSubblockMergeCand}$ estará en el intervalo de 0 a 5, inclusive.

La FIGURA 12A y la FIGURA 12B muestran descripciones de la cuarta implementación.

35

S1201: Para detalles de S1201, consúltese S901. En esta memoria no se describen de nuevo los detalles. Realizar S1202 o S1206.

40

S1202: Cuando el primer indicador indica que el modo candidato usado para interpretar el bloque a procesar comprende el modo afín, analizar sintácticamente un segundo indicador, un tercer indicador y un cuarto indicador del flujo de bits, donde el segundo indicador se usa para indicar (o determinar) una longitud máxima de una primera lista de vectores de movimiento candidatos, y la primera lista de vectores de movimiento candidatos es una lista de vectores de movimiento candidatos construido para el bloque a procesar usando un modo de predicción de fusión de subbloque. Realizar S1203, S1204 o S1205.

45

S1203: Cuando el tercer indicador indica que el modo de predicción de vector de movimiento temporal avanzada está presente en el modo de predicción de fusión de subbloque, y el cuarto indicador indica que el modo de predicción de vector de movimiento plano no está presente en el modo de predicción de fusión de subbloque, determinar un segundo número en función del cuarto indicador, y determinar la longitud máxima de la primera lista de vectores de movimiento candidatos en función del segundo indicador y el segundo número.

50

S1204: Cuando el tercer indicador indica que el modo de predicción de vector de movimiento temporal avanzada no está presente en el modo de predicción de fusión de subbloque, y el cuarto indicador indica que el modo de predicción de vector de movimiento plano está presente en el modo de predicción de fusión de subbloque, determinar un primer número en función del tercer indicador, y determinar la longitud máxima de la primera lista de vectores de movimiento candidatos en función del segundo indicador y el primer número.

55

S1205: Cuando el tercer indicador indica que el modo de predicción de vector de movimiento temporal avanzada no está presente en el modo de predicción de fusión de subbloque, y el cuarto indicador indica que el modo de predicción de vector de movimiento plano no está presente en el modo de predicción de fusión de subbloque, determinar el primer número en función del tercer indicador, determinar un segundo número en función del cuarto indicador, y determinar la longitud máxima de la primera lista de vectores de movimiento candidatos en función del segundo indicador, el primer número, y el segundo número.

60

Cuando el cuarto indicador indica que el modo plano no está presente en el modo de predicción de fusión de subbloque, el segundo número puede ser igual al número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo plano. Por ejemplo, el cuarto indicador puede ser representado por

65

sps_planar_enabled_flag. Cuando sps_planar_enabled flag = 0, indica que el modo de predicción de vector de movimiento plano no está presente en el modo de predicción de fusión de subbloque. En este caso, el segundo número es igual al número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo ATMVP. Por el contrario, cuando sps_planar_enabled flag = 1, indica que el modo de predicción de vector de movimiento plano está presente en el modo de predicción de fusión de subbloque. En este caso, el segundo número es igual a 0. Por ejemplo, el número máximo de vectores de movimiento que son soportados en predicción realizada usando el modo plano puede ser 1. En este caso, el segundo número puede ser igual a un valor de sps_planar_enabled_flag.

En un ejemplo posible, cuando el primer indicador indica que el modo candidato usado para interpredicir el bloque a procesar comprende el modo afín, la longitud máxima de la primera lista de vectores de movimiento candidatos puede obtenerse según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = K - K_minus_max_num_subblock_merge_cand - L1 - L2,$$

Por ejemplo, en la cuarta implementación, puede permitirse que un valor de K_minus_max_num_subblock_merge_cand esté en un intervalo de 0 a 2 o 0 a 3.

Por ejemplo, puede permitirse que un valor máximo de MaxNumSubblockMergeCand sea 5 o 6.

Cuando se permite que el valor máximo de MaxNumSubblockMergeCand sea 5, el segundo indicador puede representarse por five_minus_max_num_subblock_merge_cand. Cuando se permite que el valor máximo de MaxNumSubblockMergeCand sea 6, el segundo indicador puede representarse por six_minus_max_num_subblock_merge_cand.

En un ejemplo, L1 puede obtenerse según la siguiente fórmula:

$$L1 = \text{sps_sbtmvp_enabled_flag} == 1 ? 0 : 1. \text{ If } \text{sps_sbtmvp_enabled_flag} = 1, L1 = 0. \text{ If}$$

$$\text{sps_sbtmvp_enabled_flag} = 0, L1 = 1.$$

En un ejemplo, L2 puede obtenerse según la siguiente fórmula:

$$L2 = \text{sps_planar_enabled_flag} == 1 ? 0 : 1. \text{ If } \text{sps_planar_enabled_flag} = 1, L2 = 0. \text{ If}$$

$$\text{sps_sbtmvp_enabled_flag} = 0, L2 = 1.$$

S1206 a S1209: Para detalles de S1206 a S1209, consúltese de S1104 a S1107. En esta memoria no se describen de nuevo los detalles.

Por ejemplo, puede permitirse que el valor máximo de MaxNumSubblockMergeCand sea 5. Un valor de five_minus_max_num_subblock_merge_cand está en un intervalo de 0 a 2.

Si sps_affine_enable_flag = 0, MaxNumSubblockMergeCand se obtiene según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag} + \text{sps_planar_enabled_flag}.$$

De otro modo, (si sps_affine_enable_flag = 1), MaxNumSubblockMergeCand se obtiene según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_merge_cand} -$$

$$(\text{sps_sbtmvp_enabled_flag} == 1 ? 0 : 1) - (\text{sps_planar_enabled_flag} == 1 ? 0 : 1),$$

donde (five_minus_max_num_subblock_merge_cand especifica el número máximo de candidatos de predicción de vector de movimiento (MVP) de fusión basada en subbloques soportado en el sector sustraído de 5), el número máximo de candidatos de MVP de fusión basada en subbloques, MaxNumSubblockMergeCand, se deriva de la siguiente manera:

- Si sps_affine_enabled_flag es igual a 0:

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag} + \text{sps_planar_enabled_flag};$$

- De lo contrario (sps_affine_enabled_flag es igual a 1):

$$\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_merge_cand} - (\text{sps_sbtmvp_enabled_flag} == 1 ? 0 : 1) - (\text{sps_planar_enabled_flag} == 1 ? 0 : 1).$$

Un valor de MaxNumSubblockMergeCand estará en el intervalo de 0 a 5, inclusive.

- 5 Por ejemplo, puede permitirse que el valor máximo de MaxNumSubblockMergeCand sea 6. Un valor de five_minus_max_num_subblock_merge_cand está en un intervalo de 0 a 3.

Si sps_affine_enable_flag = 0, MaxNumSubblockMergeCand se obtiene según la siguiente fórmula:

10
$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag} + \text{sps_planar_enabled_flag};$$

De otro modo, (si sps_affine_enable_flag = 1), MaxNumSubblockMergeCand se obtiene según la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = 6 - \text{six_minus_max_num_subblock_merge_cand} - (\text{sps_sbtmvp_enabled_flag} == 1 ? 0 : 1) - (\text{sps_planar_enabled_flag} == 1 ? 0 : 1).$$

15 donde (six_minus_max_num_subblock_merge_cand especifica el número máximo de candidatos de predicción de vector de movimiento (MVP) de fusión basada en subbloques soportado en el sector sustraído de 6), el número máximo de candidatos de MVP de fusión basada en subbloques, MaxNumSubblockMergeCand, se deriva de la siguiente manera:

- 20 - Si sps_affine_enabled_flag es igual a 0:

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag} + \text{sps_planar_enabled_flag};$$

- 25 - De lo contrario (sps_affine_enabled_flag es igual a 1):

$$\text{MaxNumSubblockMergeCand} = 6 - \text{six_minus_max_num_subblock_merge_cand} - (\text{sps_sbtmvp_enabled_flag} == 1 ? 0 : 1) - (\text{sps_planar_enabled_flag} == 1 ? 0 : 1).$$

30 Un valor de MaxNumSubblockMergeCand estará en el intervalo de 0 a 6, inclusive.

35 Para el segundo caso, el modo de predicción de vector de movimiento plano no se considera que está presente en el modo de fusión de subbloque. En otras palabras, cuando el modo de fusión de subbloque puede comprender al menos uno del modo afín y el modo de predicción de vector de movimiento temporal avanzada, se puede usar la primera implementación o la segunda implementación. En esta memoria no se describen de nuevo detalles.

40 En función de un mismo concepto inventivo que las realizaciones de método, una realización de esta solicitud proporciona además un aparato. Como se muestra en la FIGURA 13, el aparato 1300 puede ser específicamente un procesador en un decodificador de vídeo, un chip, un sistema de chip o un módulo en un decodificador de vídeo, por ejemplo, la unidad de decodificación por entropía 304 y/o la unidad de interpredicción 344.

45 Por ejemplo, el aparato puede comprender una unidad de análisis sintáctico 1301 y una unidad de determinación 1302. La unidad de análisis sintáctico 1301 y la unidad de determinación 1302 realizan las etapas de método descritas en las realizaciones correspondientes de la FIGURA 9 a la FIGURA 12A y la FIGURA 12B. Por ejemplo, la unidad de análisis sintáctico 1301 se puede configurar para analizar sintácticamente indicadores (por ejemplo, un primer indicador, un segundo indicador, un tercer indicador o un cuarto indicador) comprendido en un flujo de bits, y la unidad de determinación 1302 se configura para determinar una longitud máxima de una primera lista de vectores de movimiento candidatos.

50 Una realización de esta solicitud proporciona además otra estructura de un aparato usado en un decodificador. Como se muestra en la FIGURA 14, el aparato 1400 puede comprender una interfaz de comunicaciones 1410 y un procesador 1420. Opcionalmente, el aparato 1400 puede comprender además una memoria 1430. La memoria 1430 se puede disponer dentro o fuera de la aparato. Ambas de la unidad de análisis sintáctico 1301 y la unidad de determinación 1302 mostradas en la FIGURA 13 pueden ser implementadas por el procesador 1420. El procesador 1420 envía o recibe un flujo de vídeo o un flujo de bits a través de la interfaz de comunicaciones 1410, y se configura para implementar los métodos de la FIGURA 9 a la FIGURA 12A y la FIGURA 12B. En un proceso de implementación, etapas en un procedimiento de procesamiento pueden realizarse usando un circuito lógico integrado de hardware en el procesador 1420 o una instrucción en forma de software, para completar los métodos de la FIGURA 9 a la FIGURA 12A y la FIGURA 12B.

La interfaz de comunicaciones 1410 en esta realización de esta solicitud puede ser un circuito, un bus, un transceptor, o cualquier otro aparato que se pueda configurar para intercambiar información. Por ejemplo, el otro aparato puede ser un dispositivo conectado al aparato 1400. Por ejemplo, si el aparato es un codificador de vídeo, el otro aparato puede ser un decodificador de vídeo.

5 En esta realización de esta solicitud, el procesador 1420 puede ser un procesador de finalidad general, un procesador de señales digitales, un circuito integrado de aplicación específica, una matriz de puertas programables en campo u otro dispositivo lógico programable, un compuerta discreta o dispositivo lógico de transistores, o un componente de hardware discreto, y puede implementar o ejecutar los métodos, etapas y diagramas de bloques lógicos descritos en
10 las realizaciones de esta solicitud. El procesador de finalidad general puede ser un microprocesador, cualquier procesador convencional o algo semejante. Las etapas de los métodos divulgados con referencia a las realizaciones de esta solicitud pueden ser realizadas directamente por un procesador de hardware, o pueden ser realizadas usando una combinación de hardware en el procesador y una unidad de software. Código de programa ejecutado por el
15 procesador 1420 para implementar los métodos anteriores puede almacenarse en la memoria 1430. La memoria 1430 se acopla al procesador 1420.

El acoplamiento en esta realización de esta solicitud puede ser un acoplamiento indirecto o una conexión de comunicación entre aparatos, unidades, o módulos en forma eléctrica, forma mecánica, u otra forma, y se usa para intercambio de información entre los aparatos, las unidades o los módulos.

20 El procesador 1420 puede funcionar en colaboración con la memoria 1430. La memoria 1430 puede ser una memoria no volátil, por ejemplo, una unidad de disco duro (HDD) o un disco de estado sólido (SSD), o puede ser una memoria volátil, por ejemplo, una memoria de acceso aleatorio (RAM). La memoria 1430 es cualquier otro medio que se puede configurar para llevar o almacenar código de programa esperado en forma de instrucción o una estructura de datos y a la que puede acceder un ordenador, pero no se limita a esto.

En esta realización de esta solicitud, un medio de conexión específico entre la interfaz de comunicaciones 1410, el procesador 1420 y la memoria 1430 no está limitado. En esta realización de esta solicitud, la memoria 1430, el procesador 1420, y la interfaz de comunicaciones 1410 se conectan a través de un bus en la FIGURA 14. El bus se
30 representa por una línea gruesa en la FIGURA 14. Un modo de conexión entre otros componentes se describe meramente de manera esquemática, y no se limita al mismo. El bus puede clasificarse en un bus de dirección, un bus de datos, un bus de control y similares. Por facilidad de la representación, en la FIGURA 14 únicamente se usa una línea gruesa para representar el bus, pero esto no significa que únicamente haya un bus o únicamente un tipo de bus.

35 Las implementaciones factibles y las realizaciones específicas anteriores relacionadas con la FIGURA 9 a la FIGURA 12A y la FIGURA 12B proporcionar descripciones de uno o más aparatos de decodificación de datos de vídeo en esta solicitud. Debe entenderse que, según las descripciones anteriores, un lado de codificador usualmente determina un modo de interpredicción y codifica el modo de interpredicción en un flujo de bits. Tras seleccionarse finalmente un modo de interpredicción, un indicador (por ejemplo, el primer indicador, el segundo indicador, el tercer indicador, o el cuarto indicador descrito anteriormente) para el modo de interpredicción se codifica en el flujo de bits usando un
40 proceso de codificación esto es completamente inverso al método de decodificación anterior (el proceso de codificación corresponde al proceso de descodificación de analizar sintácticamente el primer indicador, el segundo indicador, el tercer indicador o el cuarto indicador). Debe entenderse que determinar una longitud máxima de una primera lista de vectores de movimiento candidatos por el lado de codificador está completamente en consonancia con la del lado de decodificador. No se describe una realización específica del lado de codificador. Sin embargo, debe entenderse que el método de predicción de imágenes de vídeo descrito en esta solicitud también se usó en un aparato de codificación.

Una realización de esta solicitud proporciona además un aparato usado en un codificador. Como se muestra en la FIGURA 15, el aparato 1500 puede comprender una interfaz de comunicaciones 1510 y un procesador 1520. Opcionalmente, el aparato 1500 puede comprender además una memoria 1530. La memoria 1530 se puede disponer dentro o fuera de la aparato. El procesador 1520 envía o recibe un flujo de vídeo o un flujo de bits a través de la interfaz de comunicaciones 1510.

55 En un aspecto, el procesador 1520 se configura para codificar un primer indicador en un flujo de bits; y cuando el primer indicador indica que un modo candidato usado para interprededir el bloque a procesar comprende un modo afín, codificar un segundo indicador en el flujo de bits, donde el segundo indicador se usa para indicar una longitud máxima de una primera lista de vectores de movimiento candidatos, y la primera lista de vectores de movimiento candidatos es una lista de vectores de movimiento candidatos construido para el bloque a procesar usando un modo de predicción de fusión de subbloque.

60 La interfaz de comunicaciones 1510 en esta realización de esta solicitud puede ser un circuito, un bus, un transceptor, o cualquier otro aparato que se pueda configurar para intercambiar información. Por ejemplo, el otro aparato puede ser un dispositivo conectado al aparato 1500. Por ejemplo, si el aparato es un codificador de vídeo, el otro aparato puede ser un decodificador de vídeo.

En esta realización de esta solicitud, el procesador 1520 puede ser un procesador de finalidad general, un procesador de señales digitales, un circuito integrado de aplicación específica, una matriz de puertas programables en campo u otro dispositivo lógico programable, un compuerta discreta o dispositivo lógico de transistores, o un componente de hardware discreto, y puede implementar o ejecutar los métodos, etapas y diagramas de bloques lógicos descritos en las realizaciones de esta solicitud. El procesador de finalidad general puede ser un microprocesador, cualquier procesador convencional o algo semejante. Las etapas de los métodos divulgados con referencia a las realizaciones de esta solicitud pueden ser realizadas directamente por un procesador de hardware, o pueden ser realizadas usando una combinación de hardware en el procesador y una unidad de software. Código de programa ejecutado por el procesador 1520 para implementar los métodos anteriores puede almacenarse en la memoria 1530. La memoria 1530 se acopla al procesador 1520.

El acoplamiento en esta realización de esta solicitud puede ser un acoplamiento indirecto o una conexión de comunicación entre aparatos, unidades, o módulos en forma eléctrica, forma mecánica, u otra forma, y se usa para intercambio de información entre los aparatos, las unidades o los módulos.

El procesador 1520 puede funcionar en colaboración con la memoria 1530. La memoria 1530 puede ser una memoria no volátil, por ejemplo, una unidad de disco duro (HDD) o un disco de estado sólido (SSD), o puede ser una memoria volátil, por ejemplo, una memoria de acceso aleatorio (RAM). La memoria 1530 es cualquier otro medio que se puede configurar para llevar o almacenar código de programa esperado en forma de instrucción o una estructura de datos y a la que puede acceder un ordenador, pero no se limita a esto.

En esta realización de esta solicitud, un medio de conexión específico entre la interfaz de comunicaciones 1510, el procesador 1520 y la memoria 1530 no está limitado. En esta realización de esta solicitud, la memoria 1530, el procesador 1520 y la interfaz de comunicaciones 1510 se conectan a través de un bus en la FIGURA 15. El bus se representa por una línea gruesa en la FIGURA 15. Un modo de conexión entre otros componentes se describe meramente de manera esquemática, y no se limita al mismo. El bus puede clasificarse en un bus de dirección, un bus de datos, un bus de control y similares. Por facilidad de la representación, en la FIGURA 15 únicamente se usa una línea gruesa para representar el bus, pero esto no significa que únicamente haya un bus o únicamente un tipo de bus.

Sobre la base de las realizaciones anteriores, una realización de esta solicitud proporciona además un soporte de almacenamiento informático. El soporte de almacenamiento almacena un programa de software. Cuando el programa de software es leído y ejecutado por uno o más procesadores, el método proporcionado en una cualquiera o más de las realizaciones anteriores se puede implementar. El soporte de almacenamiento informático puede comprender cualquier soporte que pueda almacenar código de programa, tal como una unidad de memoria USB, un disco duro extraíble, una memoria de solo lectura, una memoria de acceso aleatorio, un disco magnético o un disco óptico.

Sobre la base de las realizaciones anteriores, una realización de esta solicitud proporciona además un chip. El chip comprende un procesador, configurado para implementar las funciones en una cualquiera o más de las realizaciones anteriores, por ejemplo, obtener o procesar información o un mensaje usado en los métodos anteriores. Opcionalmente, el chip comprende además una memoria. La memoria se configura para almacenar una instrucción de programa y datos que son necesarios y ejecutados por el procesador. El chip puede comprender un chip, o puede comprender un chip y otro dispositivo discreto.

Aunque se han descrito aspectos específicos de esta solicitud con referencia al codificador de vídeo 20 y el decodificador de vídeo 30, debe entenderse que las tecnologías de esta solicitud pueden ser usadas por muchas otras unidades de codificación y/o decodificación de vídeo, procesadores, unidades de procesamiento, y unidades de decodificación basadas en hardware y similares, por ejemplo, codificadores/decodificadores (CÓDEC). Adicionalmente, debe entenderse que las etapas descritas y mostradas en la FIGURA 8A a la FIGURA 12A y la FIGURA 12B se proporcionan meramente como implementaciones factibles. En otras palabras, las etapas descritas en las implementaciones factibles en la FIGURA 8A a la FIGURA 12A y la FIGURA 12B no se realizan necesariamente en el orden mostrado de la FIGURA 8A a la FIGURA 12A y la FIGURA 12B, y se pueden realizar menos etapas, adicionales o alternativas.

Además, debe entenderse que dependiendo de las implementaciones factibles, acciones o acontecimientos específicos en cualquiera de los métodos descritos en esta memoria descriptiva pueden realizarse en órdenes diferentes, puede añadirse un acción o un acontecimiento, o las acciones o acontecimientos pueden combinarse, u omitirse (por ejemplo, no todas las acciones o acontecimientos descritos son necesarios para implementar los métodos). Además, en una implementación factible particular, las acciones o acontecimientos pueden experimentar (por ejemplo) procesamiento multihilo o procesamiento de interrupción, o pueden ser procesadas por una pluralidad de procesadores simultáneamente en lugar de secuencialmente. Además, aunque aspectos específicos de esta solicitud se describen como realizados por un solo módulo o unidad con el propósito de claridad, debe entenderse que las tecnologías de esta solicitud pueden ser realizadas por una combinación de unidades o módulos asociados con el decodificador de vídeo.

En una o más implementaciones factibles, las funciones descritas pueden implementarse usando hardware, software, firmware, o cualquier combinación de los mismos. Si las funciones se implementan usando software, las funciones

pueden almacenarse en medios legibles por ordenador, como una o más instrucciones o código, o transmitirse a través de un medio legible por ordenador y ejecutarse por una unidad de procesamiento basada en hardware. El medio legible por ordenador puede comprender un soporte de almacenamiento legible por ordenador o un medio de comunicaciones. El soporte de almacenamiento legible por ordenador corresponde a un medio tangible tal como un soporte de almacenamiento de datos. El medio de comunicaciones comprende cualquier medio que facilite transmisión de un programa informático (por ejemplo) desde una ubicación a otra ubicación según un protocolo de comunicaciones.

De esta manera, el medio legible por ordenador puede corresponder a, por ejemplo, (1) un soporte de almacenamiento tangible no transitorio legible por ordenador o (2) un medio de comunicación tal como una señal o una portadora. El soporte de almacenamiento de datos puede ser cualquier soporte disponible al que pueda 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 comprender un medio legible por ordenador.

A modo de implementación factible en lugar de limitación, el soporte de almacenamiento legible por ordenador puede comprender una RAM, una ROM, una EEPROM, un CD-ROM u otro aparato de almacenamiento en disco óptico, un aparato de almacenamiento en disco magnético u otro aparato de almacenamiento magnético, una memoria flash, o cualquier otro medio que se pueda usar para almacenar código requerido en forma de instrucción o una estructura de datos y al que pueda acceder un ordenador. De manera semejante, cualquier conexión puede denominarse apropiadamente medio legible por ordenador. Por ejemplo, si se transmite una instrucción desde un sitio web, un servidor u otra fuente remota a través de un cable coaxial, una fibra óptica/cable, un par trenzado, una línea digital de abonado (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, el cable coaxial, fibra óptica/cable, el par trenzado, DSL o tecnologías inalámbricas tales como infrarrojos, radio y microondas se comprenden en la definición de medio.

Sin embargo, debe entenderse que el soporte de almacenamiento legible por ordenador y el soporte de almacenamiento de datos pueden no comprender una conexión, una portadora, una señal, u otro medio transitorio, sino que alternativamente significa soportes de almacenamiento tangibles no transitorios. Un disco magnético y un disco óptico descritos en esta memoria descriptiva comprenden un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), y un disco Blu-ray. El disco magnético usualmente reproduce datos magnéticamente, mientras que los discos reproducen datos ópticamente mediante un láser. Una combinación del disco magnético y disco óptico anteriores también estará comprendida en un alcance del medio legible por ordenador.

Una instrucción puede ser ejecutada por uno o más procesadores tales como uno o más procesadores de señales digitales (DSP), microprocesadores de finalidad general, circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), u otros circuitos integrados o lógicos discretos equivalentes. Por lo tanto, el término "procesador" usado en esta memoria descriptiva puede ser uno cualquiera de las estructuras anteriores u otra estructura que se use para implementar las tecnologías descritas en esta memoria descriptiva. Adicionalmente, en algunos aspectos, las funciones descritas en esta memoria descriptiva se pueden proporcionar dentro de módulos dedicados hardware y/o software configurados para codificar y descodificar, o puede incorporarse en un códec combinado. De manera semejante, las tecnologías pueden implementarse en uno o más circuitos o elementos lógicos.

Las tecnologías en esta solicitud pueden implementarse en diversos aparatos o dispositivos, incluido un telefonía móvil, un circuito integrado (IC), o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta solicitud para enfatizar aspectos funcionales de un aparato configurado para realizar las tecnologías descritas, pero se implementan necesariamente por diferentes unidades de hardware. Más precisamente, como se ha descrito anteriormente, se pueden combinar diversas unidades en una unidad de hardware de códec o proporcionarse mediante unidades de hardware interoperativas (incluyendo uno o más procesadores como se ha descrito anteriormente) en combinación con un conjunto apropiado de software y/o firmware.

REIVINDICACIONES

1. Un método de predicción de imágenes de vídeo, que comprende:

5 analizar, de un conjunto de parámetros de secuencia, SPS, en un flujo de bits, un valor de un primer indicador `sps_affine_enabled_flag` y un valor de un tercer indicador `sps_sbtmvp_enabled_flag`; en donde, `sps_affine_enabled_flag` se usa para especificar si un modelo de movimiento afín se puede usar para
 10 interpretar en un bloque de imagen comprendido en una imagen de vídeo; y `sps_sbtmvp_enabled_flag` se usa para especificar si un modo de predicción de vector de movimiento temporal avanzada, ATMVP, se puede usar para interpretar un bloque de imagen comprendido en una imagen de vídeo;
 cuando el valor del primer indicador es igual a 1, analizar un valor de un segundo indicador `five_minus_max_num_subblock_merge_cand` del SPS,

15 determinar una longitud máxima de una primera lista de vectores de movimiento candidatos `MaxNumSubblockMergeCand`, en donde un valor de `MaxNumSubblockMergeCand` satisface la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_merge_cand},$$

20 en donde la primera lista de vectores de movimiento candidatos es una lista de vectores de movimiento candidatos construida para un bloque a procesar, un modo de predicción de fusión de subbloque se usa para el bloque a procesar; y

25 cuando el valor del primer indicador es igual a 0, determinar la longitud máxima de la primera lista de vectores de movimiento candidatos `MaxNumSubblockMergeCand`, en donde, el valor de `MaxNumSubblockMergeCand` satisface la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag}.$$

30 2. El método de la reivindicación 1, en donde determinar una longitud máxima de una primera lista de vectores de movimiento candidatos en función del valor del tercer indicador comprende:
 determinar la longitud máxima de la primera lista de vectores de movimiento candidatos es 0, cuando el valor del tercer indicador es 0.

35 3. El método de la reivindicación 1 o 2, en donde determinar una longitud máxima de una primera lista de vectores de movimiento candidatos en función del valor del tercer indicador comprende:
 determinar la longitud máxima de la primera lista de vectores de movimiento candidatos es 1, cuando el valor del tercer indicador es 1.

40 4. El método de la reivindicación 3, en donde el valor del tercer indicador es 1 indica que un modo ATMVP está presente en el modo de predicción de fusión de subbloque.

45 5. El método de la reivindicación 2, en donde el valor del tercer indicador es 0 indica que un modo ATMVP no está presente en el modo de predicción de fusión de subbloque.

6. El método de la reivindicación 4 o 5, en donde el modo ATMVP es un modo basado en función de vector de movimiento temporal de subbloque.

50 7. El método de cualquiera de las reivindicaciones 1 a 6, en donde el valor del primer indicador indica si un modo candidato usado para interpretar el bloque a procesar comprende un modo afín.

8. Un aparato de predicción de imágenes de vídeo, que comprende:

55 una unidad de análisis sintáctico, configurada para analizar, de un conjunto de parámetros de secuencia, SPS, en un flujo de bits, un valor de un primer indicador `sps_affine_enabled_flag` y un valor de un tercer indicador `sps_sbtmvp_enabled_flag`; en donde, el `sps_affine_enabled_flag` se usa para especificar si un modelo de movimiento afín se puede usar para interpretar en un bloque de imagen comprendido en una imagen de vídeo; y el `sps_sbtmvp_enabled_flag` se usa para especificar si un modo de predicción de vector de movimiento temporal avanzada, ATMVP, se puede usar para interpretar un bloque de imagen comprendido en una
 60 imagen de vídeo;

la unidad de análisis sintáctico se configura para analizar sintácticamente un valor de un segundo indicador `five_minus_max_num_subblock_merge_cand` del SPS, cuando el valor del primer indicador es igual a 1;
 una unidad de determinación, configurada para determinar la longitud máxima de la primera lista de vectores de movimiento candidatos `MaxNumSubblockMergeCand` cuando el valor del primer indicador es igual a 1, en
 65 donde un valor de `MaxNumSubblockMergeCand` satisface la siguiente fórmula:

$$\text{MaxNumSubblockMergeCand} = 5 - \text{five_minus_max_num_subblock_merge_cand},$$

5 en donde la primera lista de vectores de movimiento candidatos es una lista de vectores de movimiento candidatos construido para un bloque a procesar, un modo de predicción de fusión de subbloque se usa para el bloque a procesar;
la unidad de determinación, se configura para determinar la longitud máxima de la primera lista de vectores de movimiento candidatos `MaxNumSubblockMergeCand` cuando el valor del primer indicador es igual a 0, en donde el valor de `MaxNumSubblockMergeCand` satisface la siguiente fórmula:

10

$$\text{MaxNumSubblockMergeCand} = \text{sps_sbtmvp_enabled_flag}.$$

9. El aparato de la reivindicación 8, en donde determinar una longitud máxima de una primera lista de vectores de movimiento candidatos en función del valor del tercer indicador comprende:
15 determinar la longitud máxima de la primera lista de vectores de movimiento candidatos es 0, cuando el valor del tercer indicador es 0.

10. El aparato de la reivindicación 8 o 9, en donde determinar una longitud máxima de una primera lista de vectores de movimiento candidatos en función del valor del tercer indicador comprende:
20 determinar la longitud máxima de la primera lista de vectores de movimiento candidatos es 1, cuando el valor del tercer indicador es 1.

11. El aparato de la reivindicación 10, en donde el valor del tercer indicador es 1 indica que un modo ATMVP está presente en el modo de predicción de fusión de subbloque.

25 12. El aparato de la reivindicación 9, en donde el valor del tercer indicador es 0 indica que un modo ATMVP no está presente en el modo de predicción de fusión de subbloque.

30 13. El aparato de la reivindicación 11 ó 12, en donde el modo ATMVP es un modo basado en función de vector de movimiento temporal de subbloque.

14. El aparato de cualquiera de las reivindicaciones 8 a 13, en donde el valor del primer indicador indica si un modo candidato usado para interprededir el bloque a procesar comprende un modo afín.

35 15. Un producto de programa informático que comprende código de programa que, cuando se ejecuta en un ordenador o un procesador, provoca que el ordenador o el procesador realice el método según una cualquiera de las reivindicaciones 1-7.

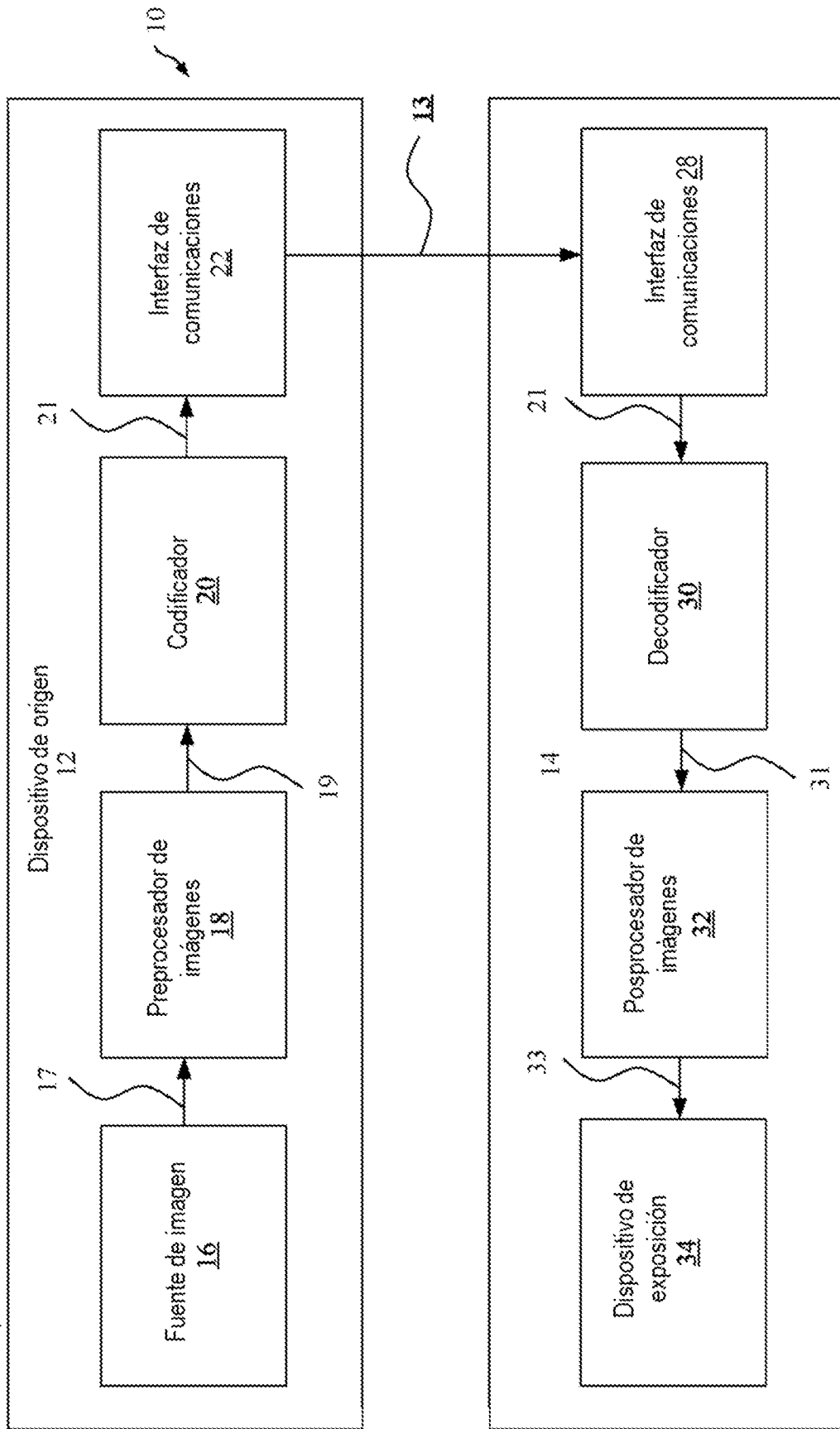


FIG. 1A

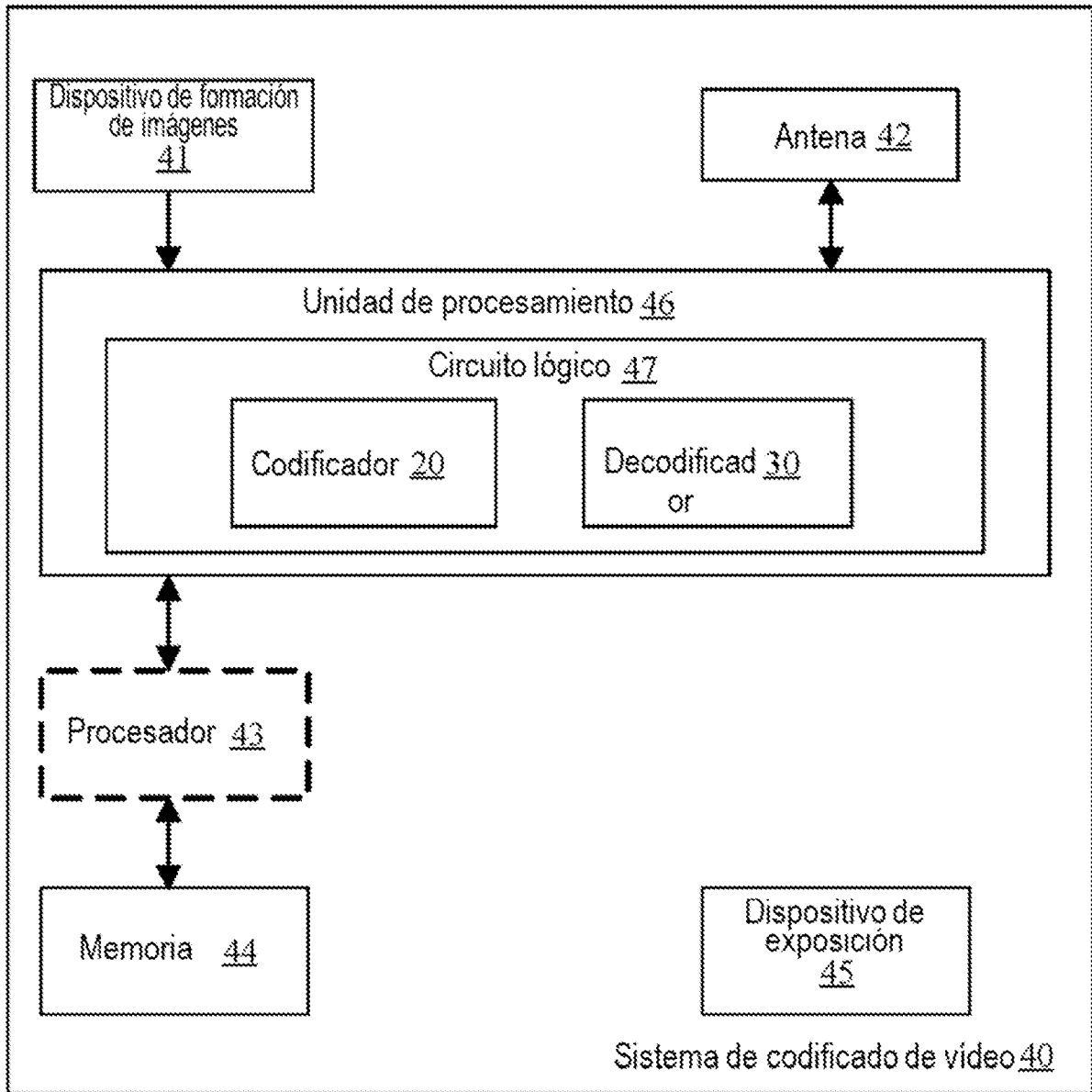


FIG. 1B

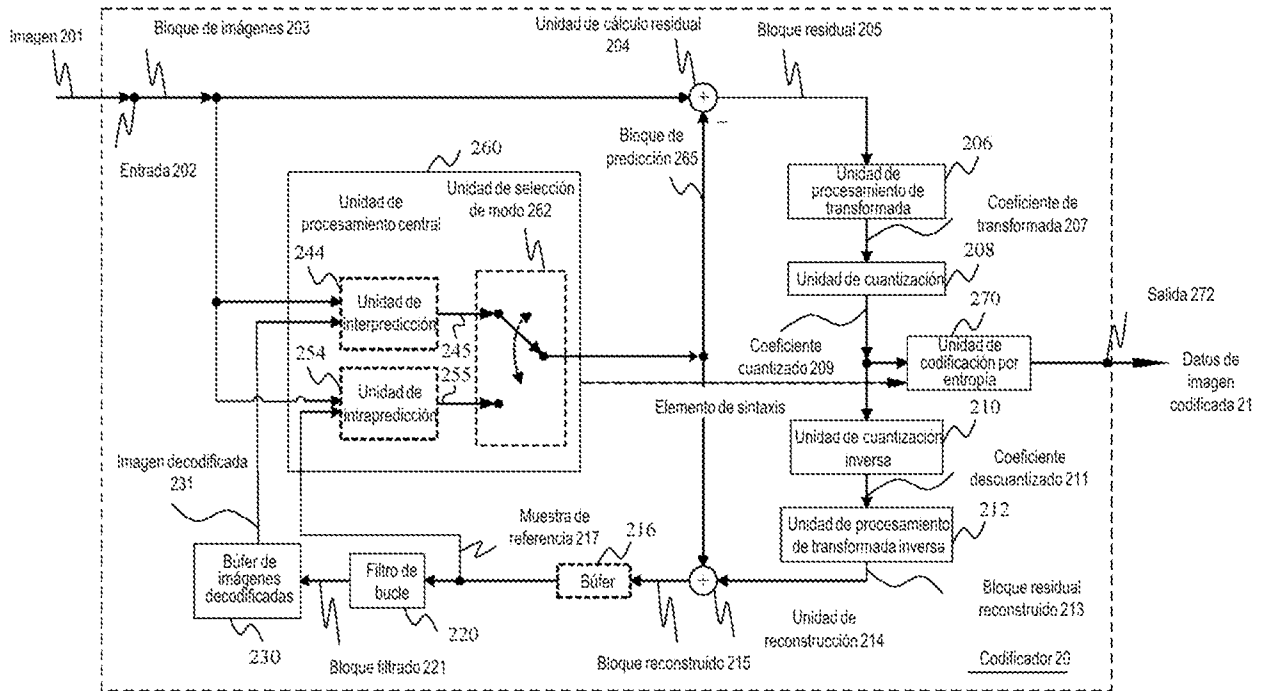


FIG. 2

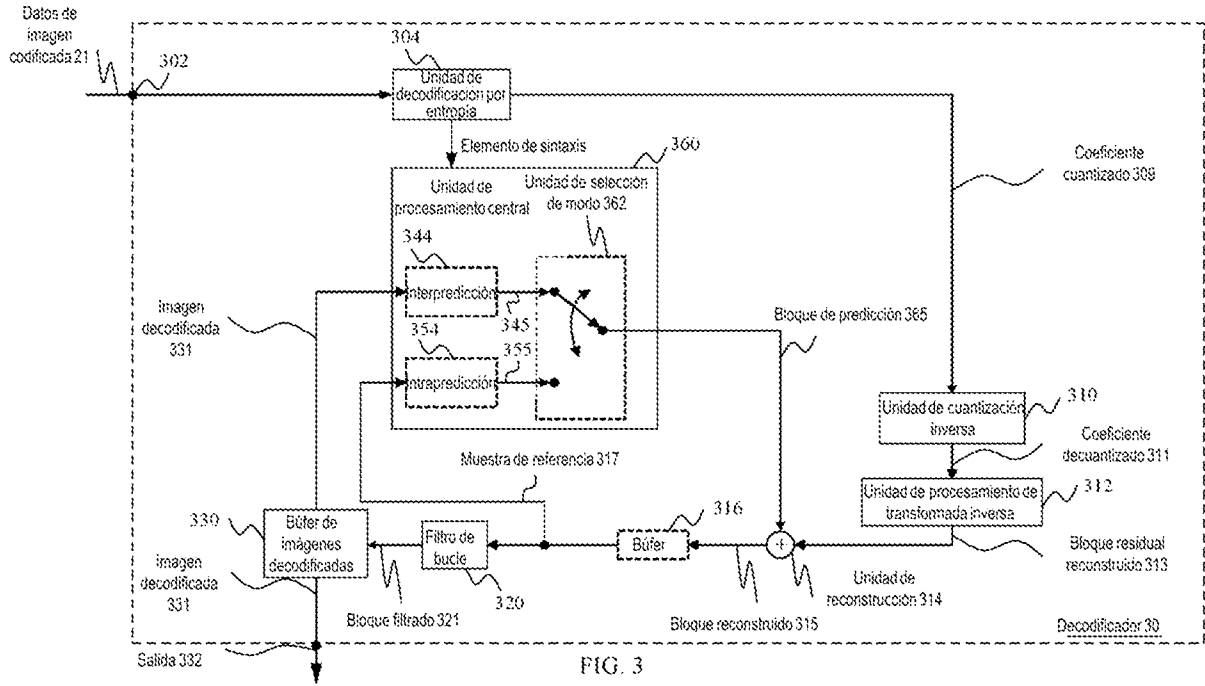


FIG. 3

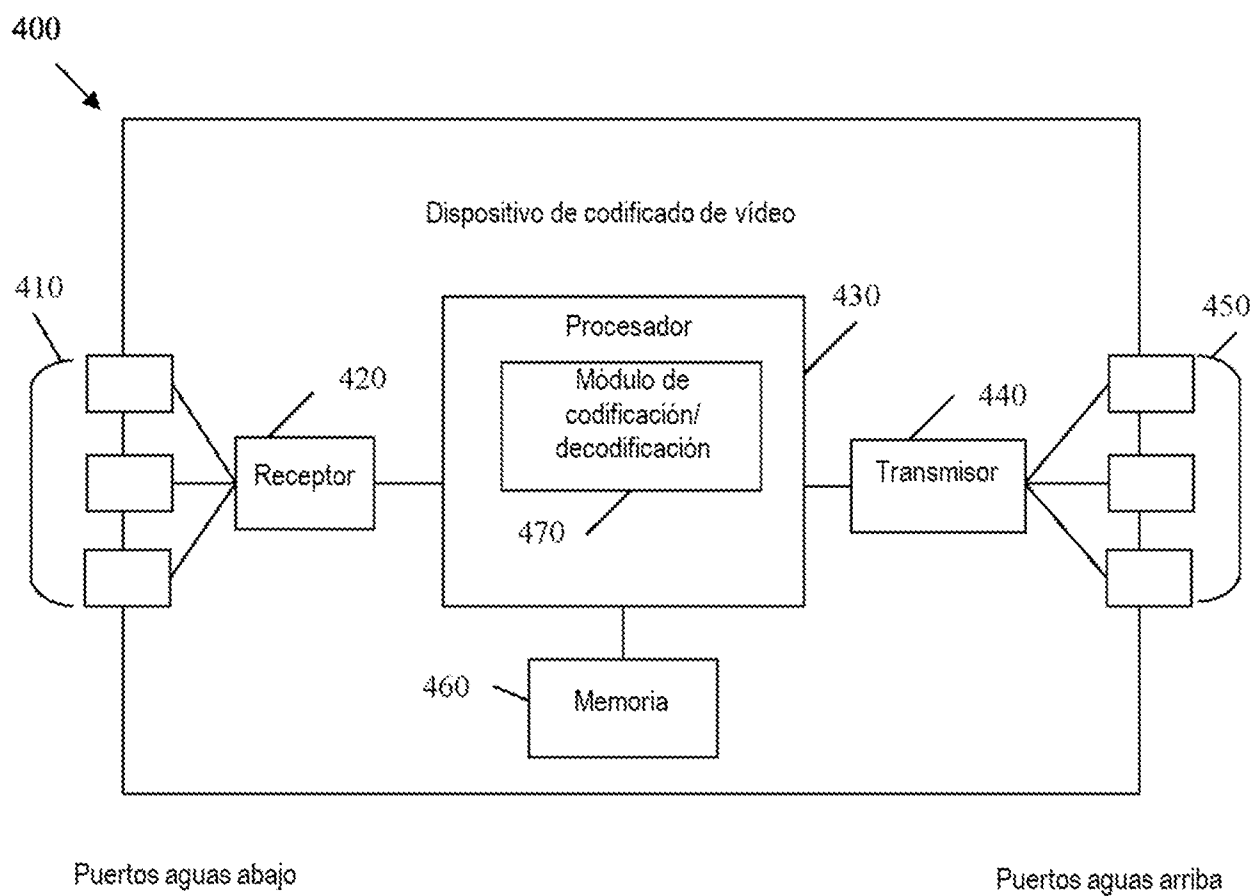


FIG. 4

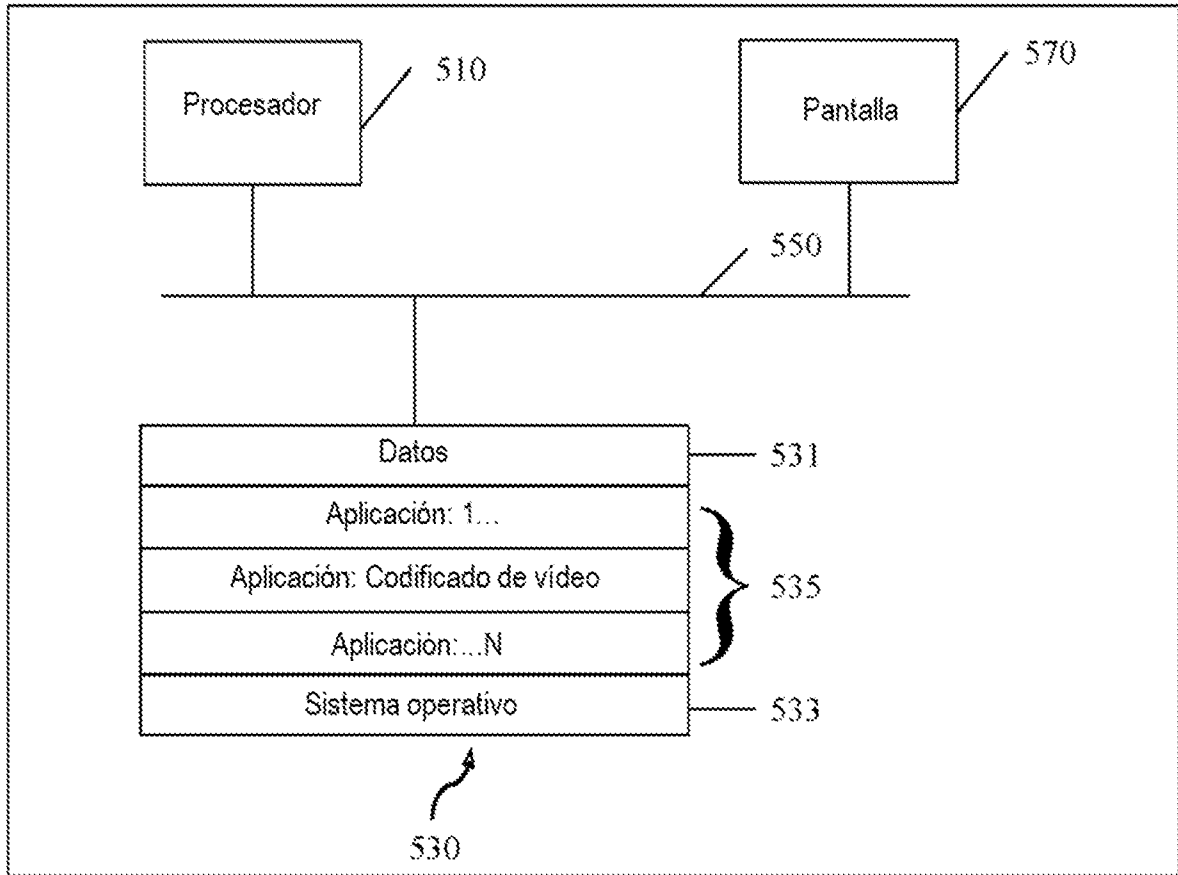


FIG. 5

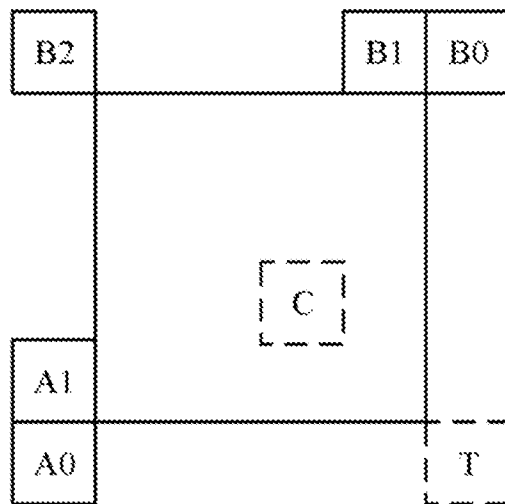


FIG. 6A

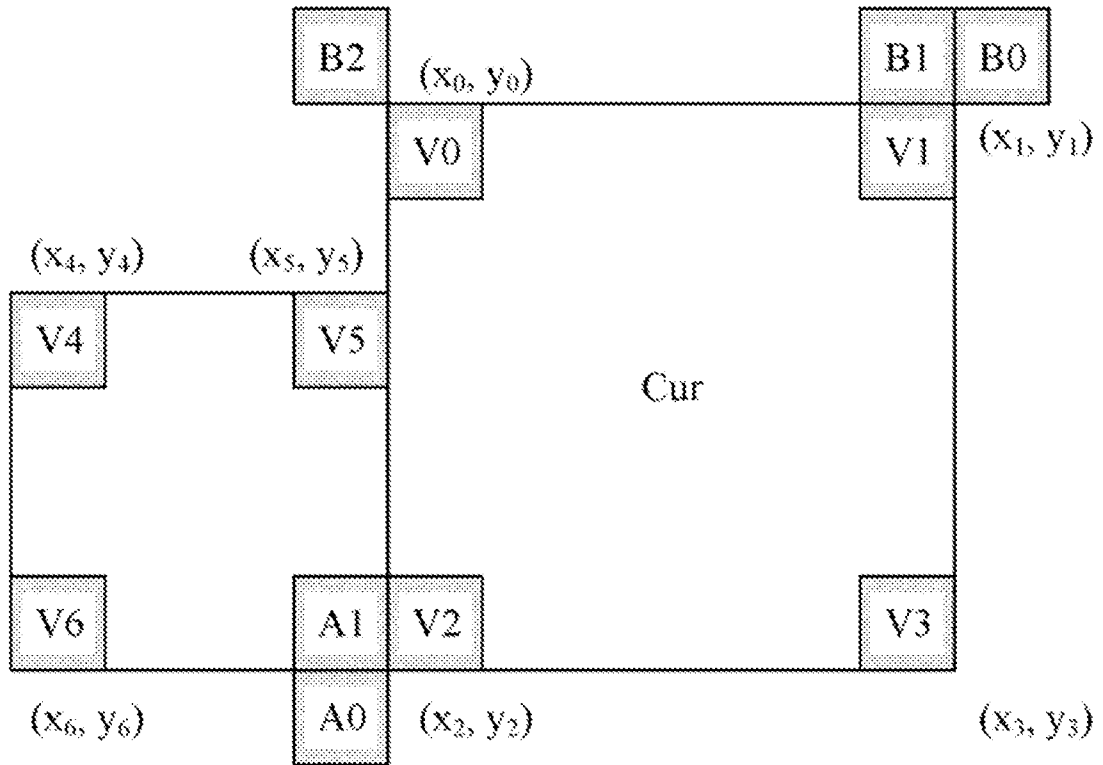


FIG. 6B

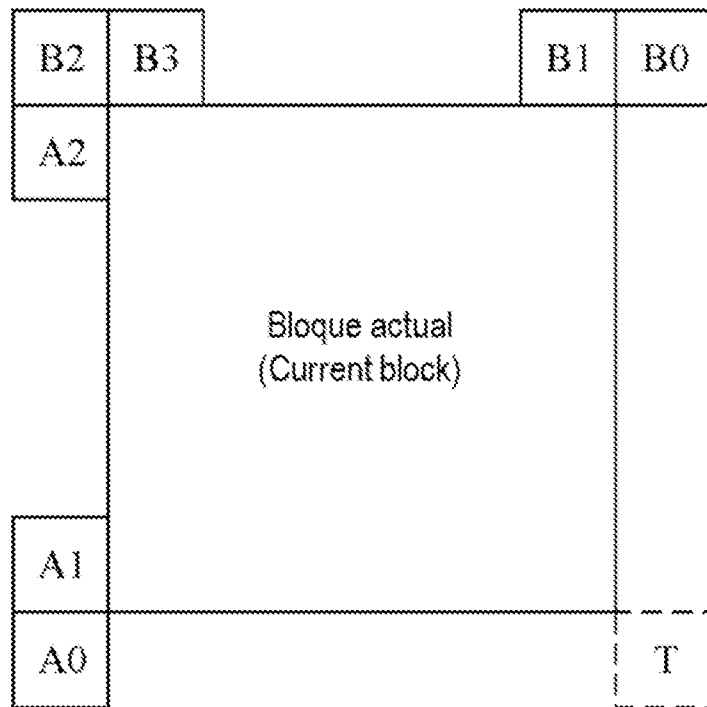


FIG. 6C

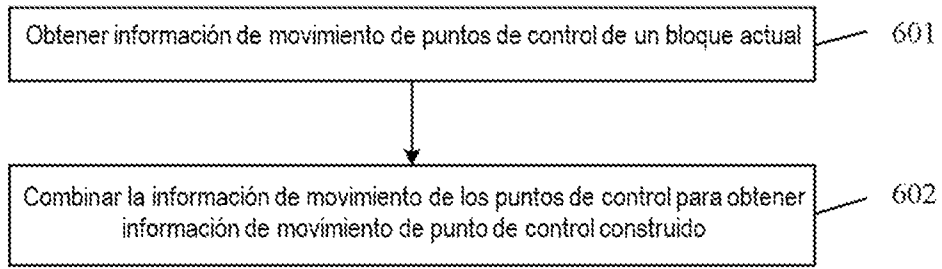


FIG. 6D

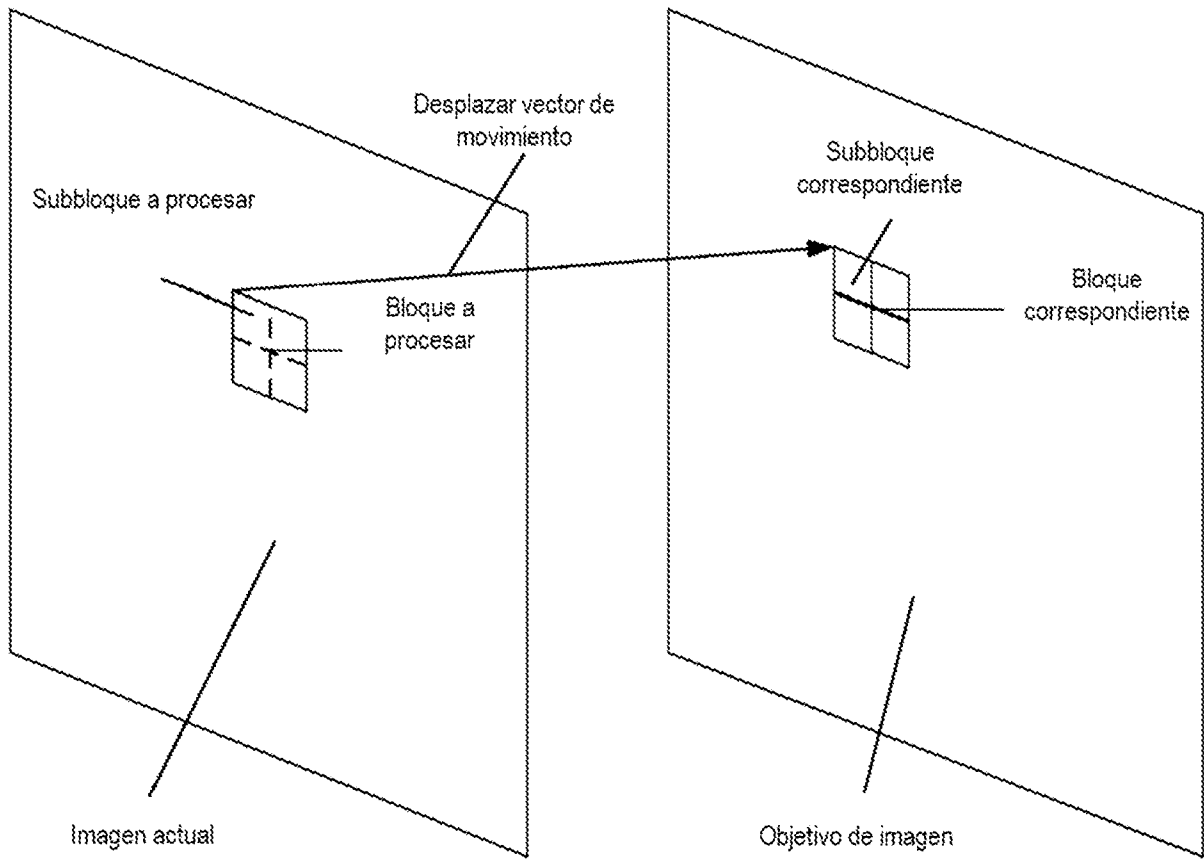


FIG. 6E

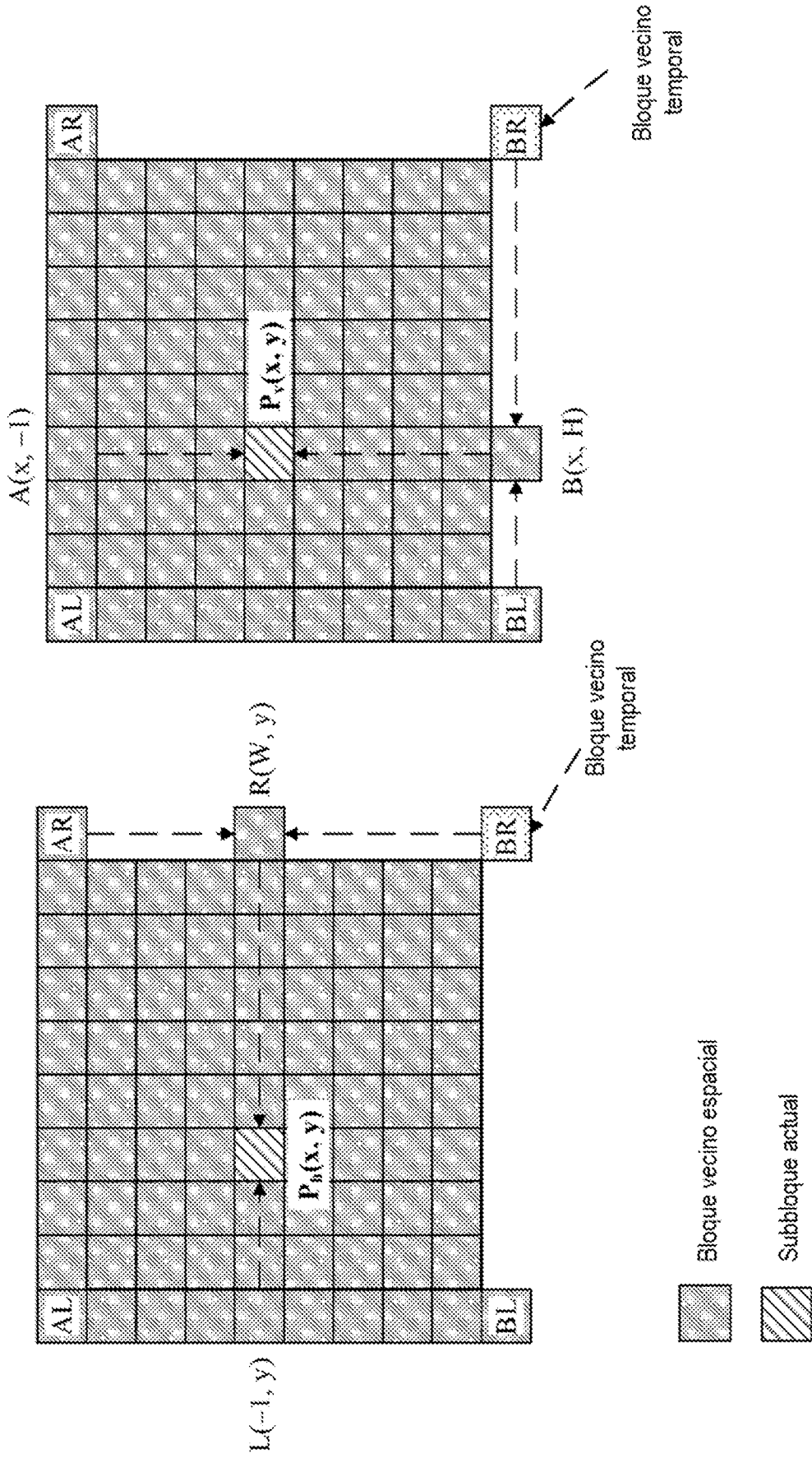


FIG. 7

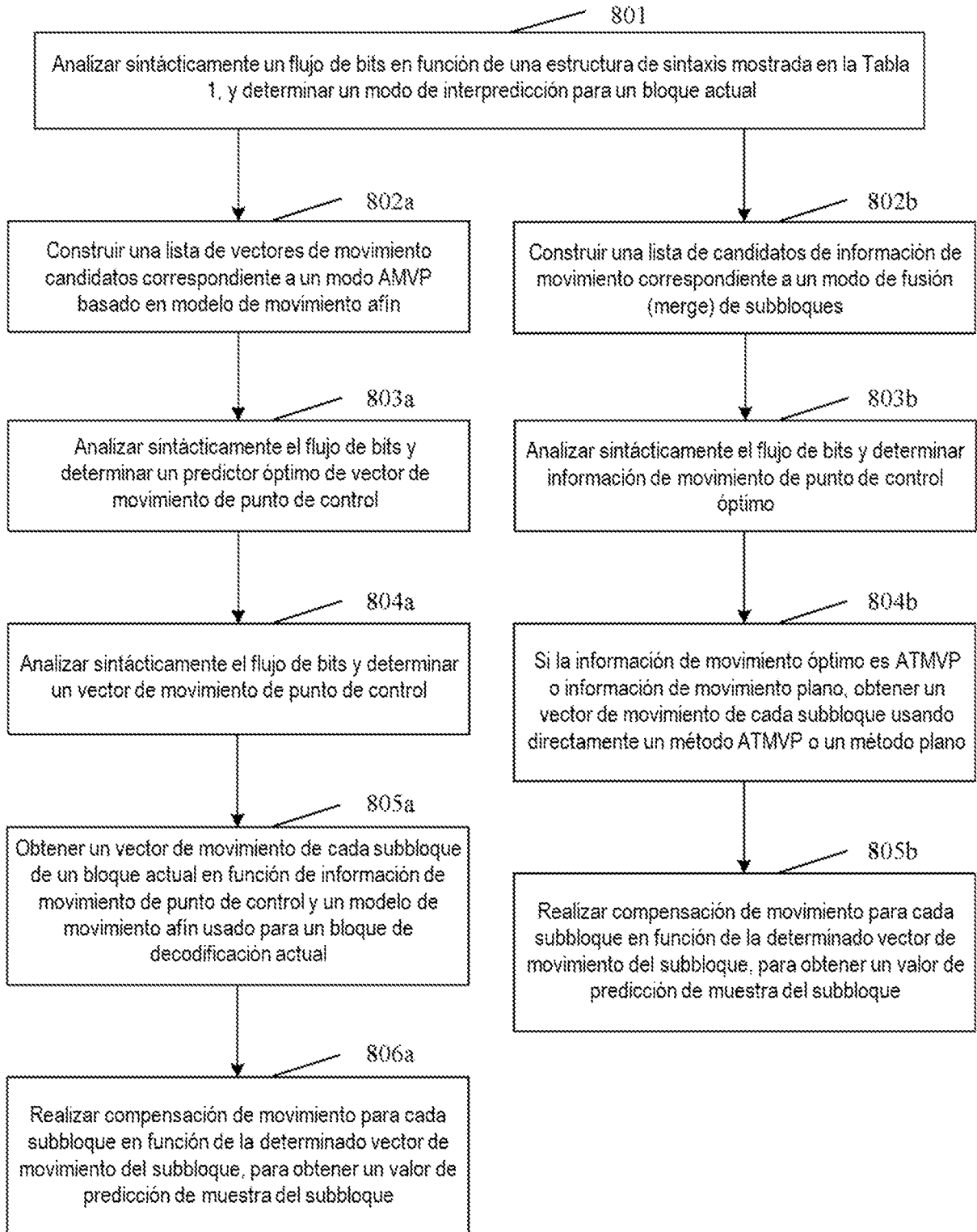


FIG. 8A

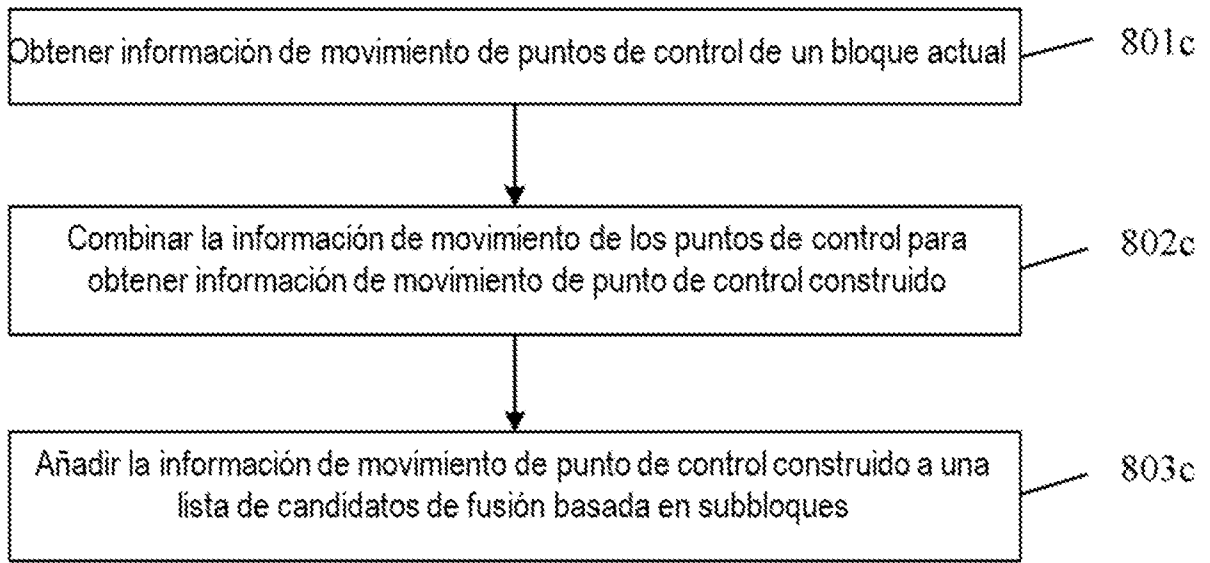


FIG. 8B

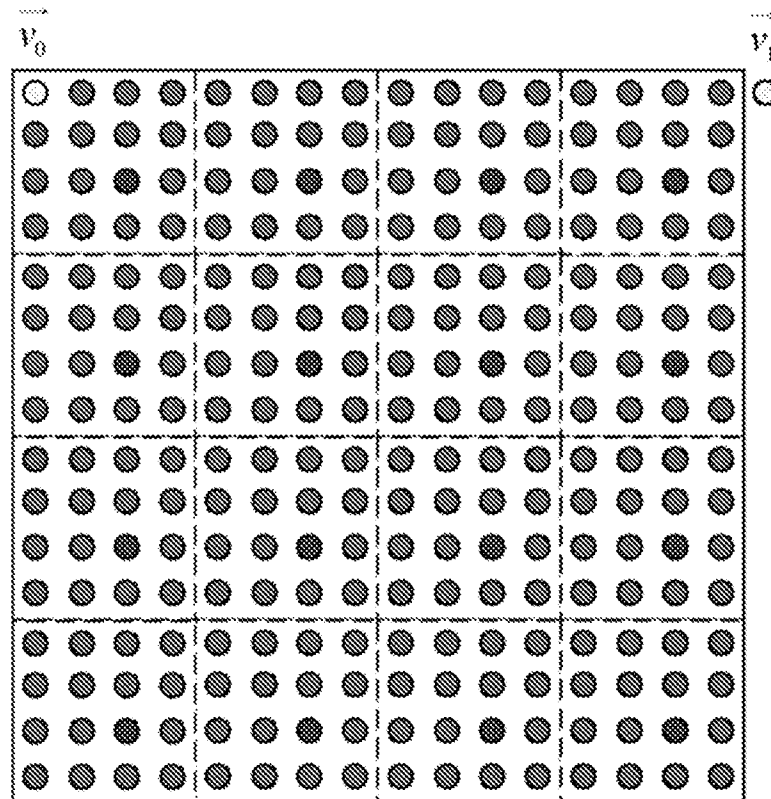


FIG. 8C

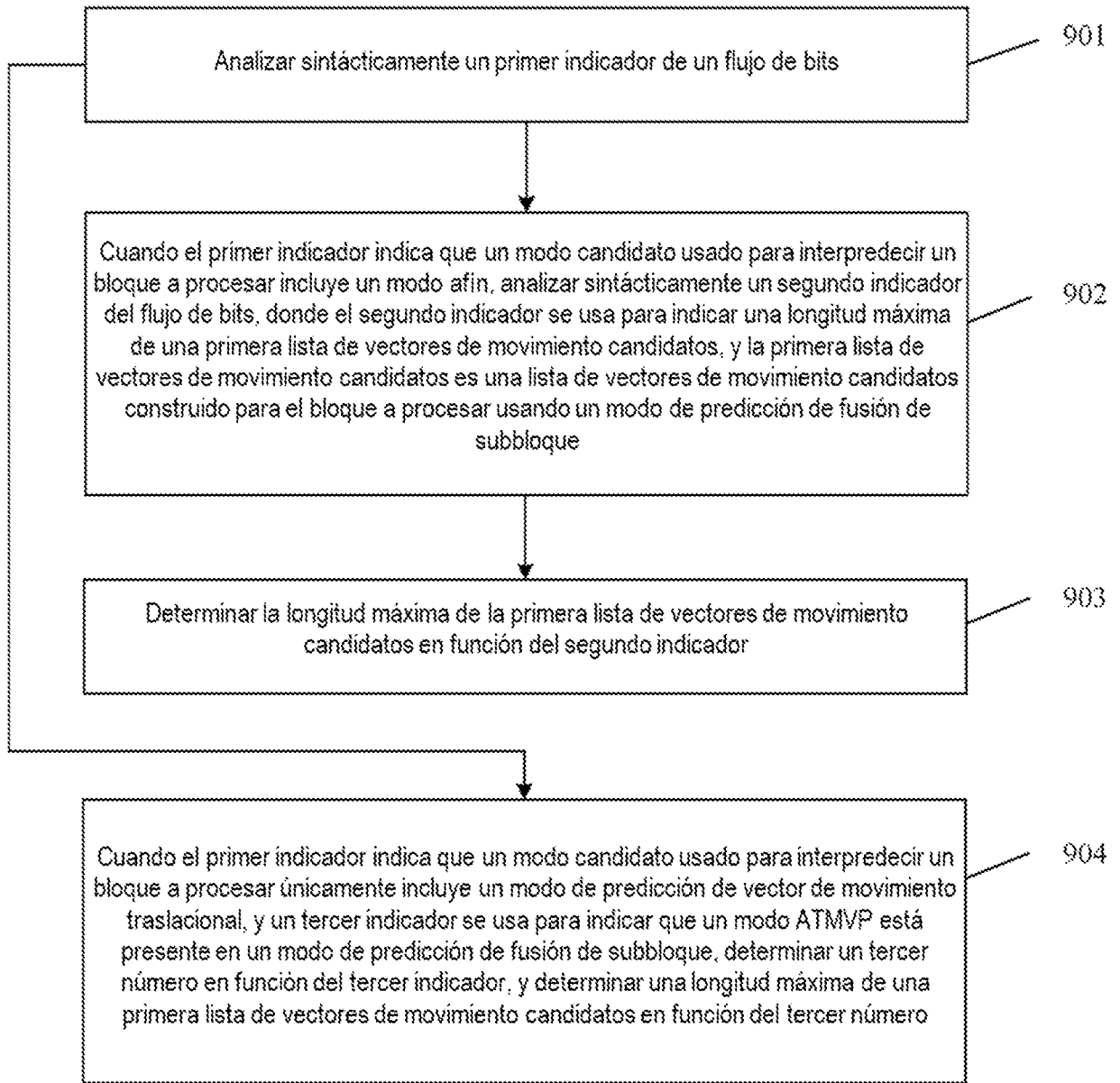


FIG. 9

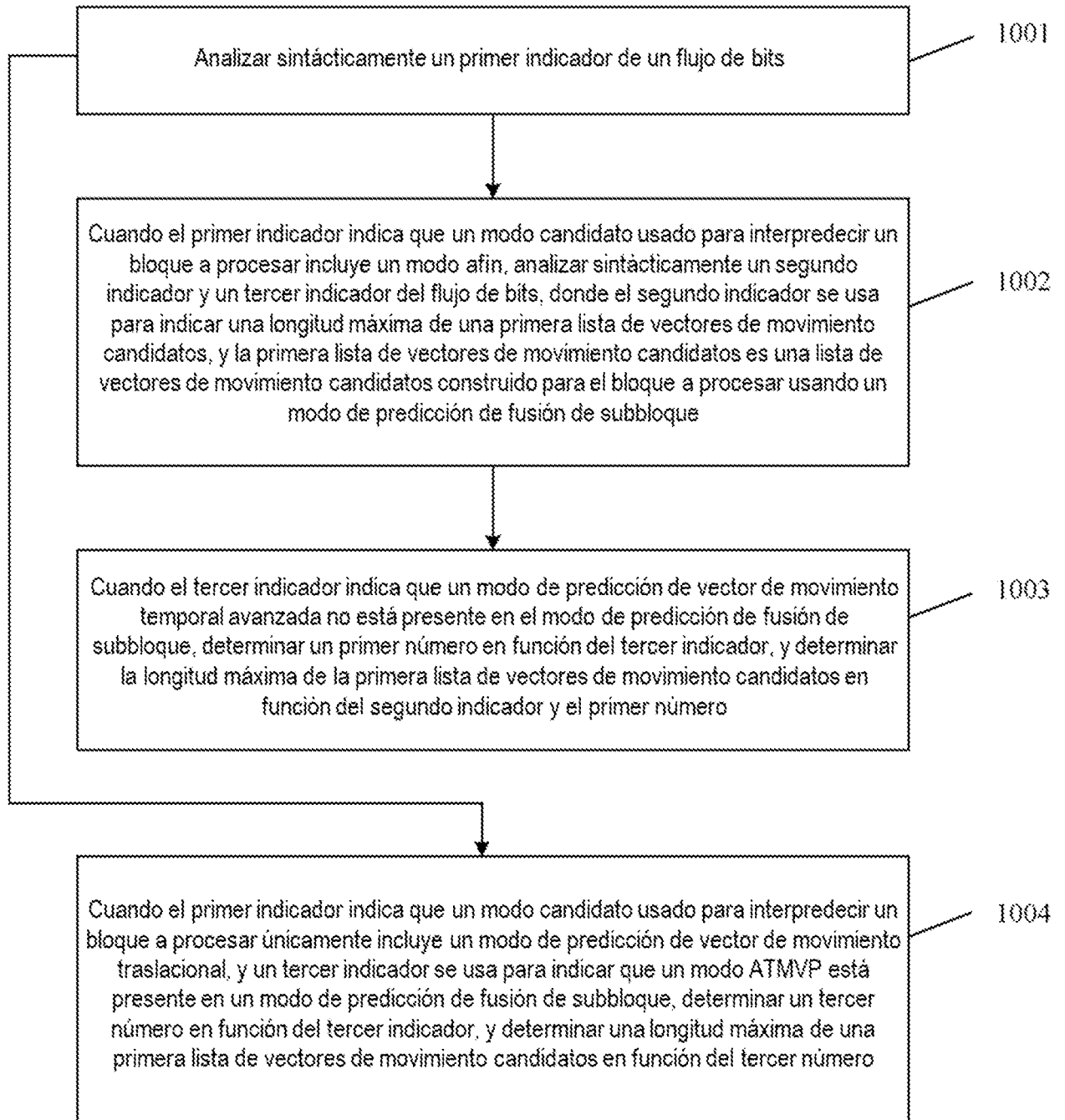


FIG. 10

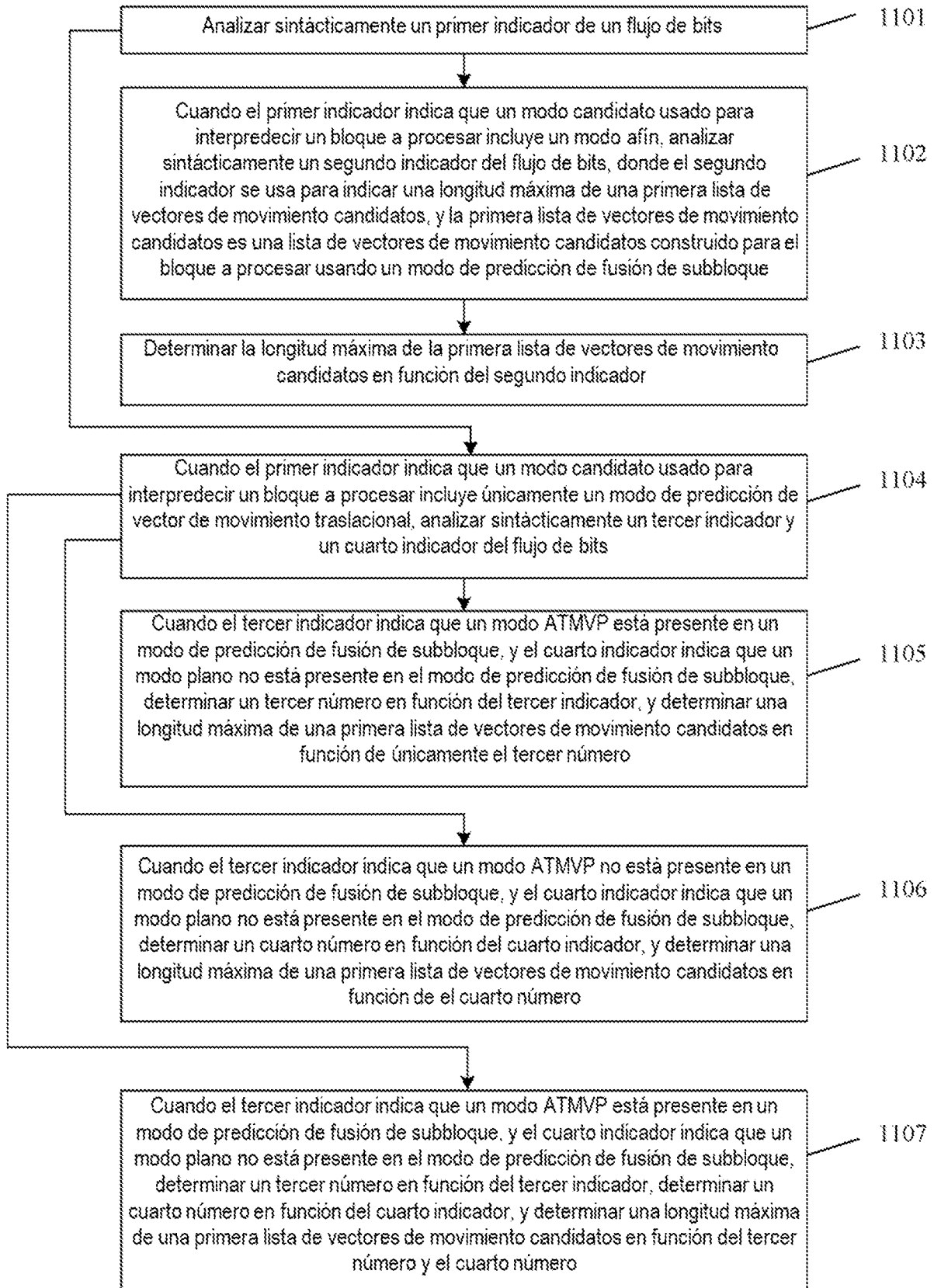
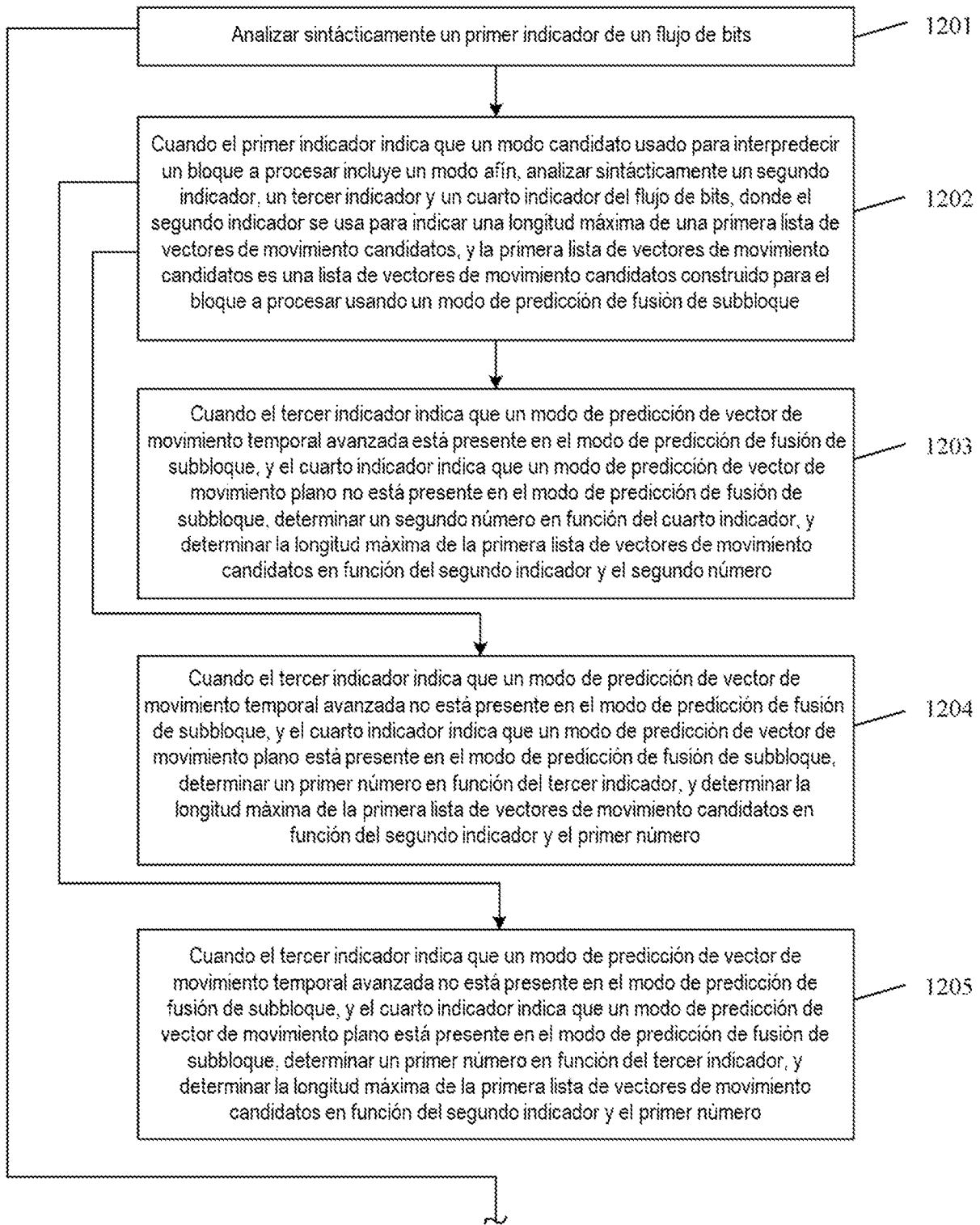


FIG. 11



A LA FIG. 12B

FIG. 12A

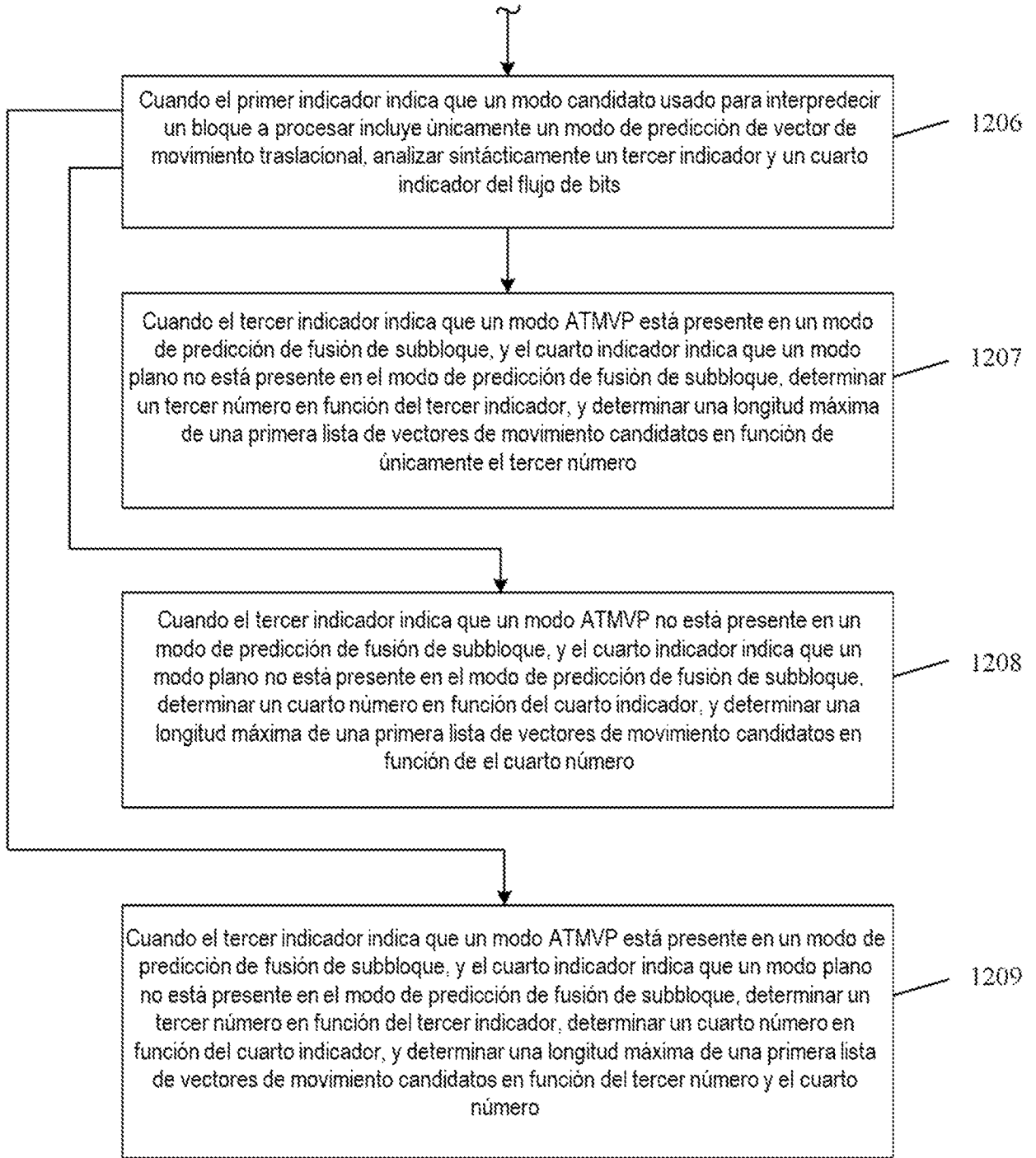


FIG. 12B

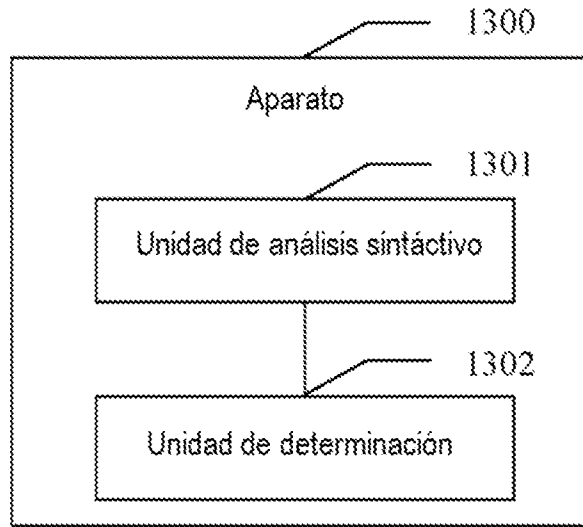


FIG. 13

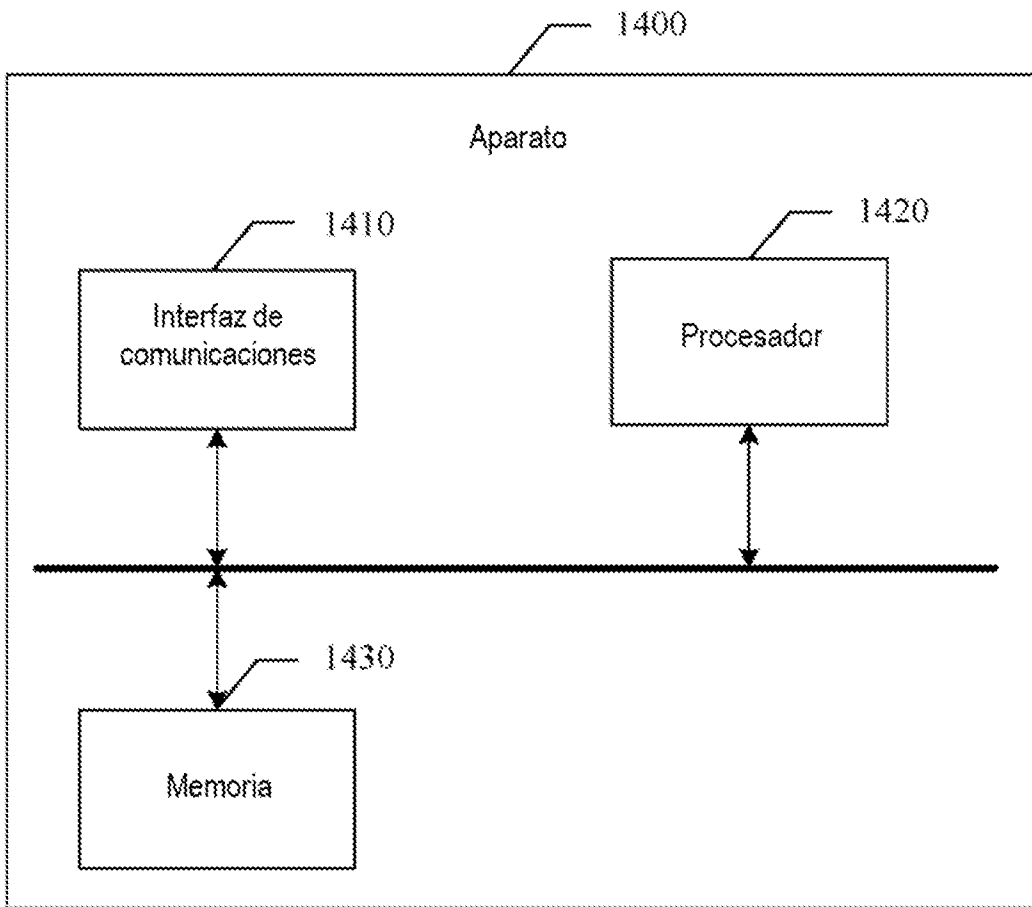


FIG. 14

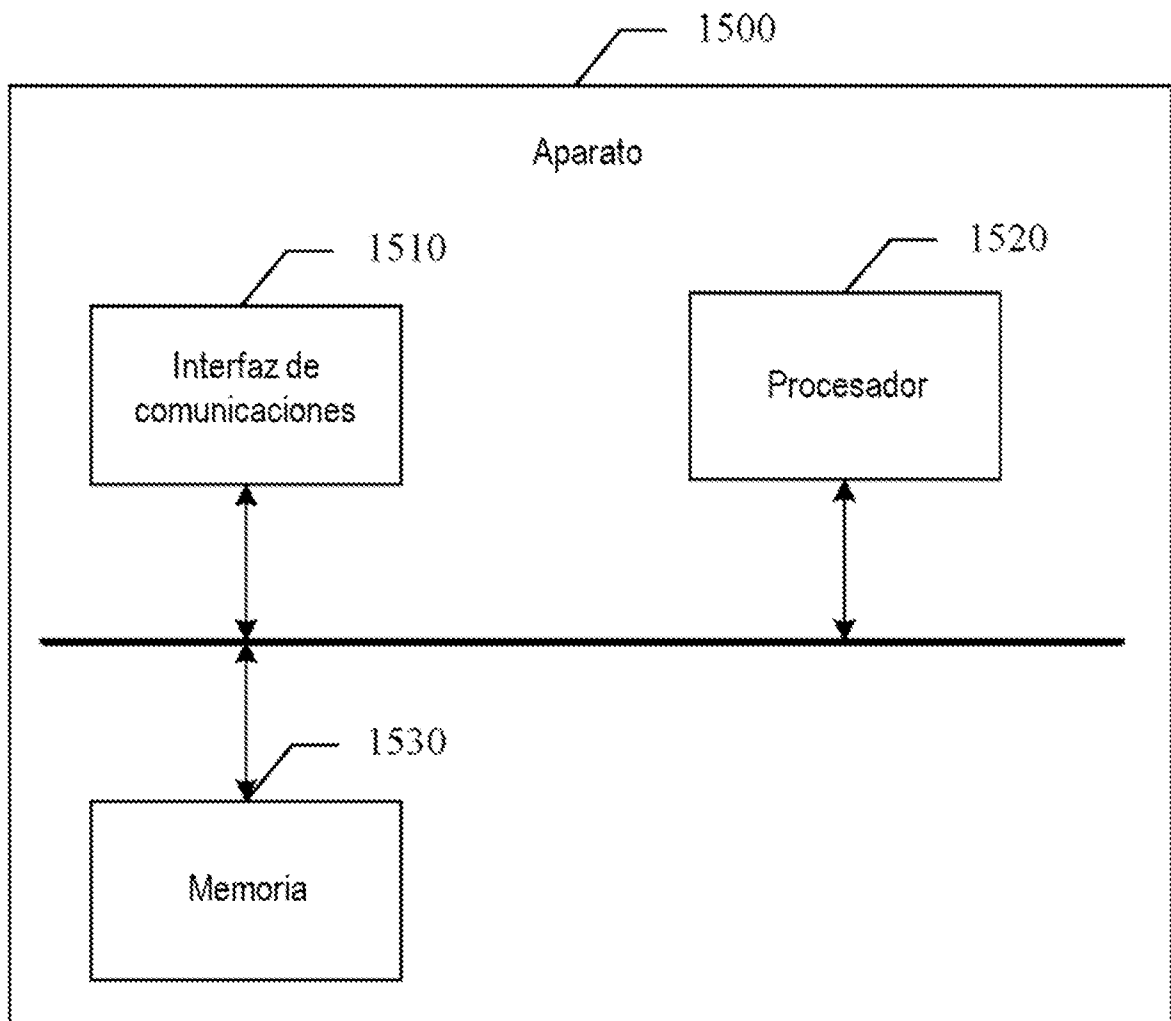


FIG. 15