

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 979 151**

51 Int. Cl.:

H04N 19/52 (2014.01)

H04N 19/537 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **05.05.2017 PCT/US2017/031258**

87 Fecha y número de publicación internacional: **23.11.2017 WO17200771**

96 Fecha de presentación y número de la solicitud europea: **05.05.2017 E 17723623 (9)**

97 Fecha y número de publicación de la concesión europea: **29.05.2024 EP 3459249**

54 Título: **Predicción de movimiento afín para la codificación de vídeo**

30 Prioridad:

16.05.2016 US 201662337301 P
04.05.2017 US 201715587044

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
24.09.2024

73 Titular/es:

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

72 Inventor/es:

ZOU, FENG;
CHEN, JIANLE;
KARCZEWICZ, MARTA;
LI, XIANG;
CHUANG, HSIAO-CHIANG y
CHIEN, WEI-JUNG

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 979 151 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

Predicción de movimiento afín para la codificación de vídeo

- 5 La presente solicitud reivindica el beneficio de la solicitud provisional de los Estados Unidos Núm. 62/337,301, presentada el 16 de mayo de 2016.

Campo técnico

- 10 Esta divulgación se refiere a la codificación de vídeo.

Antecedentes

- 15 Las capacidades de vídeo digital pueden incorporarse en una amplia gama de dispositivos, que incluye televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, celular o teléfonos por radio satélite, los llamados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo en directo, y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tal como las descritas en los estándares de codificación de vídeo. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficientemente mediante la implementación de tales técnicas de codificación de vídeo.

- 25 Algunos estándares de codificación de vídeo se definen por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), incluida sus extensiones de codificación de vídeo escalable (SVC) y codificación de vídeo multivista (MVC), ITU-T H.265, también conocida como codificación de vídeo de alta eficiencia (HEVC), y extensiones de tales estándares. Recientemente, el diseño de un nuevo estándar de codificación de vídeo, específicamente, la codificación de vídeo de alta eficiencia (HEVC), ha finalizado por el equipo de colaboración conjunta en codificación de vídeo (JCT-VC) del grupo de expertos en codificación de vídeo de la ITU-T (VCEG) y el grupo de expertos de imágenes en movimiento (MPEG) de la ISO/IEC. El último borrador de la memoria descriptiva de HEVC, en lo sucesivo HEVC WD, está disponible en itu.int/rec/T-REC-H.265-201504-S/en. El JCT-VC también está desarrollando extensiones de la gama HEVC, específicamente, HEVC-Rext. Un borrador de trabajo (WD) reciente de las extensiones de la gama, denominado en lo sucesivo RExt WD6, está disponible en phenix.int-evry.fr/jct/doc_end_user/documents/16_San%20Jose/wg11/JCTVC-P1005-v1.zip.

- 35 Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimágenes) y/o la predicción temporal (interimágenes) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo (por ejemplo, un fotograma de vídeo o una porción de un fotograma de vídeo) puede particionarse en bloques de vídeo, que en algunas técnicas también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un segmento intracodificado (I) de una imagen se codifican mediante el uso de la predicción espacial con respecto a las muestras de referencia en los bloques vecinos de la misma imagen. Los bloques de vídeo en un segmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse como fotogramas, y las imágenes de referencia pueden denominarse como fotogramas de referencia.

- 45 La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan las diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forma el bloque predictivo, y los datos residuales indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una compresión adicional, los datos residuales se pueden transformar del dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, que luego se pueden cuantificar. Los coeficientes de transformación que se cuantifican, dispuestos inicialmente en una matriz bidimensional, pueden escanearse con el fin de producir un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación de entropía para lograr una compresión aún mayor.

- 50 El documento YUWEN HE Y OTROS: "Efficient coding with adaptive motion models", publicado para el 23. PICTURE CODING SYMPOSIUM, SAINT MALO, el 23 de abril de 2003 (XP030080026) divulga un procedimiento de codificación de vídeo eficiente, en el que se explotan modelos de movimiento adaptativos en la predicción con compensación de movimiento. La diferencia entre el procedimiento propuesto y la codificación de compensación de movimiento tradicional es que no solo se usa el modelo de movimiento traslacional con dos parámetros, sino que también se utiliza el de cuatro o seis parámetros y tamaños de bloque variables. El modo seleccionado entre esas alternativas se codifica explícitamente en el encabezado del macrobloque. Los vectores de movimiento de los puntos de control de los modelos afines se predicen individualmente a partir de los bloques vecinos y las diferencias de los

vectores de movimiento se codifican en el flujo de bits.

Sumario

- 5 La invención protegida se define por la combinación de características especificadas, respectivamente, en las reivindicaciones independientes adjuntas.

A continuación, se describen varios ejemplos, aspectos y realizaciones para ayudar a comprender la invención, pero sin proporcionar una extensión de la protección más allá del ámbito de las reivindicaciones.

10 En un ejemplo, un procedimiento para decodificar datos de vídeo incluye: obtener, mediante uno o más procesadores de un decodificador de vídeo y para un bloque actual de datos de vídeo, valores de los vectores de movimiento de un modelo de movimiento afín de un bloque vecino de datos de vídeo; derivar, mediante el uno o más procesadores y a partir de los valores de los vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo, valores de predictores para los vectores de movimiento de un modelo de movimiento afín del bloque actual de datos de vídeo; decodificar, mediante el uno o más procesadores y a partir de un flujo de bits de vídeo codificado, una representación de las diferencias entre los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo y los valores de los predictores; determinar, mediante el uno o más procesadores, los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo a partir de los valores de los predictores y las diferencias decodificadas; determinar, en base a los valores determinados de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo, un bloque predictor de datos de vídeo; y reconstruir el bloque actual de datos de vídeo en base al bloque predictor de datos de vídeo.

25 En otro ejemplo, un procedimiento para codificar datos de vídeo incluye: determinar, mediante uno o más procesadores de un codificador de vídeo, valores de los vectores de movimiento de un modelo de movimiento afín de un bloque actual de datos de vídeo, los vectores de movimiento del modelo de movimiento afín que identifican un bloque predictor de datos de vídeo para el bloque actual de datos de vídeo; obtener, mediante el uno o más procesadores, valores de los vectores de movimiento de un modelo de movimiento afín de un bloque vecino de datos de vídeo; derivar, mediante el uno o más procesadores y a partir de los valores de los vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo, valores de predictores para los vectores de movimiento de un modelo de movimiento afín del bloque actual de datos de vídeo; y codificar, mediante el uno o más procesadores y en un flujo de bits de vídeo codificado, una representación de las diferencias entre los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo y los valores de los predictores.

En otro ejemplo, un dispositivo para decodificar un bloque de datos de vídeo incluye: una memoria que se configura para almacenar los datos de vídeo; y una o más unidades de procesamiento implementadas en circuitos. En este ejemplo, la una o más unidades de procesamiento se configuran para: obtener, para un bloque actual de datos de vídeo, valores de los vectores de movimiento de un modelo de movimiento afín de un bloque vecino de datos de vídeo; derivar, a partir de los valores de los vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo, valores de predictores para los vectores de movimiento de un modelo de movimiento afín del bloque actual de datos de vídeo; decodificar, a partir de un flujo de bits de vídeo codificado, una representación de las diferencias entre los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo y los valores de los predictores; determinar los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo a partir de los valores de los predictores y las diferencias decodificadas; determinar, en base a los valores determinados de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo, un bloque predictor de datos de vídeo; y reconstruir el bloque actual de datos de vídeo en base al bloque predictor de datos de vídeo.

50 En otro ejemplo, un dispositivo para codificar un bloque de datos de vídeo incluye: una memoria que se configura para almacenar los datos de vídeo; y una o más unidades de procesamiento implementadas en circuitos. En este ejemplo, la una o más unidades de procesamiento se configuran para: determinar valores de los vectores de movimiento de un modelo de movimiento afín de un bloque actual de datos de vídeo, los vectores de movimiento del modelo de movimiento afín que identifican un bloque predictor de datos de vídeo para el bloque actual de datos de vídeo; obtener valores de los vectores de movimiento de un modelo de movimiento afín de un bloque vecino de datos de vídeo; derivar, a partir de los valores de los vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo, valores de predictores para los vectores de movimiento de un modelo de movimiento afín del bloque actual de datos de vídeo; y codificar, en un flujo de bits de vídeo codificado, una representación de las diferencias entre los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo y los valores de los predictores.

65 En otro ejemplo, un dispositivo para codificar o decodificar datos de vídeo incluye: medios para obtener, para un bloque actual de datos de vídeo, valores de los vectores de movimiento de un modelo de movimiento afín de un bloque vecino de datos de vídeo; medios para derivar, a partir de los valores de los vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo, valores de predictores para los vectores de

movimiento de un modelo de movimiento afin del bloque actual de datos de vídeo; medios para obtener diferencias entre los valores de los vectores de movimiento del modelo de movimiento afin para el bloque actual de datos de vídeo y los valores de los predictores; medios para determinar cada uno de los valores de los vectores de movimiento del modelo de movimiento afin para el bloque actual de datos de vídeo a partir de los valores de los predictores y las diferencias decodificadas; y medios para identificar, en base a los valores determinados de los vectores de movimiento del modelo de movimiento afin para el bloque actual de datos de vídeo, un bloque predictor de datos de vídeo.

En otro ejemplo, un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un codificador de vídeo o un decodificador de vídeo: obtener, para un bloque actual de datos de vídeo, valores de los vectores de movimiento de un modelo de movimiento afin de un bloque vecino de datos de vídeo; derivar, a partir de los valores de los vectores de movimiento del modelo de movimiento afin del bloque vecino de datos de vídeo, valores de predictores para los vectores de movimiento de un modelo de movimiento afin del bloque actual de datos de vídeo; obtener diferencias entre los valores de los vectores de movimiento del modelo de movimiento afin para el bloque actual de datos de vídeo y los valores de los predictores; determinar cada uno de los valores de los vectores de movimiento del modelo de movimiento afin para el bloque actual de datos de vídeo a partir de los valores de los predictores y las diferencias decodificadas; e identificar, en base a los valores determinados de los vectores de movimiento del modelo de movimiento afin para el bloque actual de datos de vídeo, un bloque predictor de datos de vídeo.

Los detalles de uno o más ejemplos se establecen en las figuras acompañantes y en la descripción de más abajo. Otras características, objetivos y ventajas serán evidentes a partir de la descripción y las figuras y a partir de las reivindicaciones.

Breve descripción de las figuras

La Figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo que puede configurarse para realizar las técnicas de esta divulgación.

La Figura 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede configurarse para realizar las técnicas de esta divulgación.

La Figura 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede configurarse para realizar las técnicas de esta divulgación.

Las Figuras 4A y 4B son diagramas conceptuales que ilustran los candidatos a vecinos espaciales en codificación de vídeo de alta eficiencia (HEVC).

La Figura 5 es un diagrama conceptual que ilustra un vector de movimiento afin de dos puntos con cuatro parámetros afines.

La Figura 6 es un diagrama conceptual que ilustra un modo inter afin.

Las Figuras 7A y 7B son diagramas conceptuales que ilustran candidatos para un modo de fusión afin.

La Figura 8 es un diagrama conceptual que ilustra un modelo de movimiento afin de seis parámetros, de acuerdo con una o más técnicas de esta divulgación.

La Figura 9 es un diagrama conceptual que ilustra la evaluación del vector de movimiento afin, de acuerdo con una o más técnicas de esta divulgación.

La Figura 10 es un diagrama conceptual que ilustra la compensación de movimiento de bloques solapados (OBMC) en H.263.

Las Figuras 11A y 11B son diagramas conceptuales que ilustran la OBMC sobre HEVC.

Las Figuras 12A y 12B son diagramas conceptuales que ilustran los subbloques donde se puede aplicar la OBMC.

La Figura 13 es un diagrama de flujo que ilustra un procedimiento de ejemplo para realizar una compensación de movimiento afin mediante un codificador de vídeo (por ejemplo, durante un procedimiento de codificación de vídeo), de acuerdo con una o más técnicas de esta divulgación.

La Figura 14 es un diagrama de flujo que ilustra un procedimiento de ejemplo para realizar una compensación de movimiento afin mediante un decodificador de vídeo (por ejemplo, durante un procedimiento de codificación de vídeo), de acuerdo con una o más técnicas de esta divulgación.

Descripción detallada

En general, esta divulgación describe técnicas relacionadas con la codificación (por ejemplo, codificación o decodificación) de información de movimiento afin para un bloque de datos de vídeo. En los actuales estándares de codificación de vídeo, sólo se aplican modelos de movimiento traslacional para la predicción de compensación de movimiento (MCP). Cuando se usa un modelo de movimiento traslacional para la MCP, los codificadores de vídeo (por ejemplo, codificadores de vídeo o decodificadores de vídeo) pueden utilizar un único vector de movimiento bidimensional (MV) para un bloque actual que indica un desplazamiento entre el bloque actual de datos de vídeo y un bloque predictor correspondiente de datos de vídeo. Los MV pueden ser bidimensionales en el sentido de que cada MV puede tener un componente x que indica un desplazamiento horizontal entre el bloque actual de datos de vídeo y el bloque predictor de datos de vídeo, y un componente y que indica un desplazamiento vertical entre el bloque actual de datos de vídeo y el bloque predictor de datos de vídeo. Como se divulga en más detalle más abajo,

en los actuales estándares de codificación de vídeo, tal como HEVC, hay dos modos de interpredicción, denominados fusión (la omisión se considera un caso especial de fusión) y modos de predicción de vector de movimiento avanzada (AMVP). En el modo de fusión, el valor de un MV de un bloque actual se hereda directamente del valor de un MV candidato, que puede ser el valor de un MV de un bloque vecino del bloque actual. Por el contrario, en el modo AMVP, el valor del MV candidato puede refinarse aún más. En particular, un codificador de vídeo puede señalar un valor de una diferencia entre el valor del MV candidato y el valor del MV para el bloque actual. El valor de la diferencia puede denominarse como diferencia de vector de movimiento (MVD).

Sin embargo, existen muchos tipos de movimientos distintos a los movimientos traslacionales, tal como los movimientos de acercamiento, movimientos de alejamiento, movimientos de rotación, movimientos de perspectiva y otros movimientos irregulares. Aplicar solo el modelo de movimiento traslacional para la MCP en tales secuencias de prueba con movimientos irregulares puede afectar la precisión de la predicción y puede resultar en una baja eficiencia de codificación. Por ejemplo, usar solo el modelo de movimiento traslacional puede resultar en bloques de predicción que no coincidan tan bien con los bloques originales que se están codificando. Como resultado, el tamaño de los datos residuales (es decir, los valores que representan diferencias de píxeles entre los bloques originales a codificar y el bloque de predicción), puede aumentar, lo que puede reducir la eficiencia de la codificación.

ITU-T VCEG (Q6/16) e ISO/IEC MPEG (JTC 1/SC 29/WG 11) están estudiando la posible necesidad de estandarizar la futura tecnología de codificación de vídeo con una capacidad de compresión que supere significativamente la del actual estándar HEVC (incluidas sus extensiones actuales y las extensiones a corto plazo para la codificación de contenido de pantalla y la codificación de alto rango dinámico). Los grupos están trabajando juntos en esta actividad de exploración en un esfuerzo de colaboración conjunto conocido como equipo conjunto de exploración de vídeo (JVET) para evaluar los diseños de tecnología de compresión propuestos por sus expertos en esta área. El JVET ha lanzado un modelo de exploración conjunta (JEM) que describe las características de codificación que se encuentran bajo estudio en un modelo de prueba coordinado como posible tecnología de codificación de vídeo mejorada más allá de las capacidades de HEVC. En el JEM, se proponen modelos de movimiento afines para su aplicación a la MCP. Una descripción reciente del algoritmo del JEM, "Algorithm Description of Joint Exploration Test Model 2", equipo conjunto de exploración de vídeo (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 2da reunión: San Diego, EE.UU., 20-26 de febrero de 2016, Documento: JVET-B1001_v3 (en lo sucesivo, "JEM test model") está disponible en phenix.it-sudparis.eu/jvet/doc_end_user/documents/2_San%20Diego/wg11/JVET-B1001-v3.zip.

Cuando se usan modelos de movimiento afines para la MCP, el codificador de vídeo puede utilizar múltiples vectores de movimiento para un bloque actual que indican colectivamente una transformación afín (por ejemplo, traslación, escalado, reflexión, rotación, etc.) entre el bloque actual de datos de vídeo y un bloque predictor correspondiente de datos de vídeo. Por ejemplo, un modelo de movimiento afín puede incluir un primer vector de movimiento bidimensional que indica un desplazamiento entre una esquina superior izquierda de un bloque actual y una esquina superior izquierda del bloque predictor correspondiente, y un segundo vector de movimiento bidimensional que indica un desplazamiento entre una esquina superior derecha del bloque actual y una esquina superior derecha del bloque predictor correspondiente. Los vectores de movimiento en un modelo de movimiento afín pueden denominarse como vectores de movimiento de punto de control (CPMV) y pueden referenciarse a una localización (es decir, un punto de control) en el bloque actual. Por ejemplo, un vector de movimiento bidimensional que indica un desplazamiento entre una esquina superior izquierda de un bloque actual y una esquina superior izquierda del bloque predictor correspondiente puede denominarse como CPMV superior izquierdo del bloque actual. Como se divulga en más detalle más abajo, en el modelo de prueba JEM, hay dos modos de interpredicción, inter afín (por ejemplo, AF_INTER) y fusión afín (por ejemplo, AF_MERGE).

En el modo de fusión afín, el valor de cada CPMV de un bloque actual se deriva directamente de los CPMV de un único bloque vecino del bloque actual que se codifica mediante el uso de un modelo de movimiento afín. En otras palabras, en el modo de fusión afín, los CPMV del bloque vecino simplemente se deforman a los CPMV del bloque actual y no hay flexibilidad para cambiar o ajustar los parámetros del modelo afín. En particular, no es posible modificar los valores de los CPMV mediante el uso de las MVD.

En el modo inter afín, el valor de cada CPMV de un bloque actual se deriva individualmente, en base al valor de un MV de un bloque vecino al punto de control correspondiente y una MVD. El valor del MV en el que un CPMV se determina puede denominarse como predictor del vector de movimiento del punto de control (CPMVP). Como ejemplo, el valor del CPMV superior izquierdo de un bloque actual se puede derivar en base a un MV de uno de un bloque izquierdo, un bloque superior izquierdo o un bloque superior vecino adyacente al punto superior izquierdo del bloque actual y una MVD. Como otro ejemplo, el valor del CPMV superior derecho de un bloque actual se puede derivar en base a un MV de uno de un bloque superior derecho o un bloque superior vecino adyacente al punto superior derecho del bloque actual y una MVD.

Tanto en la HEVC como en el modelo de prueba JEM, un codificador de vídeo puede señalar la sintaxis de MVD (es decir, elementos de sintaxis que representan ese valor de la MVD) en el flujo de bits para que los MV puedan reconstruirse en el lado del decodificador. La cantidad de datos usados para señalar la sintaxis de MVD puede estar relacionada con el tamaño del valor de MVD. Por ejemplo, pueden necesitarse más datos para señalar la sintaxis de

MVD para las MVD con valores relativamente mayores en comparación con las MVD con valores relativamente menores.

Sin embargo, la técnica actual de derivar el valor de cada CPMV en base al valor de un MV de un bloque vecino del punto de control correspondiente puede presentar una o más desventajas. Como ejemplo, la técnica actual no aprovecha la correlación del modelo de movimiento afín de un bloque actual y el modelo de movimiento afín de un bloque vecino.

De acuerdo con una o más técnicas de esta divulgación, un codificador de vídeo puede determinar valores de los vectores de movimiento de un modelo de movimiento afín de un bloque actual de datos de vídeo en base a los valores de los vectores de movimiento de un modelo de movimiento afín de un bloque vecino de datos de vídeo en particular y los valores de las diferencias entre los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo y los valores de los vectores de movimiento que se derivan en base al modelo de movimiento afín del bloque vecino de datos de vídeo. Por ejemplo, un codificador de vídeo puede utilizar los CPMV del bloque vecino como los CPMVP para los CPMV del bloque actual. Como los CPMV del bloque vecino pueden correlacionarse con los CMPV del bloque actual, las diferencias (por ejemplo, las MVD) entre los predictores (por ejemplo, los CPMVP) y los vectores de movimiento (por ejemplo, los CMPV) del bloque actual pueden reducirse. De esta manera, como la cantidad de datos usados para codificar las diferencias puede ser proporcional al tamaño de la diferencia, las técnicas de esta divulgación pueden mejorar la eficiencia de la compresión de vídeo.

Se ha avanzado un modelo de movimiento afín de cuatro parámetros en Huawei Technologies Co, Ltd "Affine transform prediction for next generation video coding", el documento UIT-T SG 16 (Período de estudio 2013) Contribución 1016 (en lo sucesivo, "Contribution 1016") está disponible en itu.int/md/T13-SG16-C-1016/en. La contribución 1016 introduce un modelo afín de cuatro parámetros que se muestra más abajo en la Ecuación (1).

$$\begin{cases} v_x = ax - by + c \\ v_y = bx + ay + d \end{cases} \quad (1)$$

Donde (v_{0x} , v_{0y}) es el CPMV para la esquina superior izquierda de un bloque actual y (v_{1x} , v_{1y}) es el CPMV para la esquina superior derecha del bloque actual, el modelo de movimiento afín, también denominado como campo de vector de movimiento (MVF), se puede representar de acuerdo con la ecuación (2) más abajo.

$$\begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w}x - \frac{(v_{1y} - v_{0y})}{w}y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w}x + \frac{(v_{1x} - v_{0x})}{w}y + v_{0y} \end{cases} \quad (2)$$

El modelo afín de cuatro parámetros que se muestra arriba en la ecuación (1) puede presentar una o más desventajas. En particular, el movimiento afín de cuatro parámetros restringe los parámetros afines de los componentes x e y, lo que los fuerza a tener propiedades de escala simétrica. Sin embargo, esta restricción puede no ser cierta en el contenido de vídeo diversificado.

De acuerdo con una o más técnicas de esta divulgación, un codificador de vídeo puede utilizar selectivamente o bien un modelo de movimiento afín de cuatro parámetros o un modelo de movimiento afín de seis parámetros. Por ejemplo, un decodificador de vídeo puede determinar si un bloque actual se codifica mediante el uso del modelo de movimiento afín de cuatro parámetros que se muestra arriba en la Ecuación (1) o un modelo de movimiento afín de seis parámetros que se muestra más abajo en la Ecuación (3).

$$\begin{cases} v_x = ax - by + c \\ v_y = dx + ey + f \end{cases} \quad (3)$$

En algunos ejemplos, el decodificador de vídeo puede determinar qué modelo de movimiento afín se usa en base a una señalización explícita. Por ejemplo, el codificador de vídeo puede decodificar, a partir de un flujo de bits, un elemento de sintaxis que indica si el modelo de movimiento afín para un bloque actual de datos de vídeo comprende un modelo de cuatro parámetros o un modelo de seis parámetros. En algunos ejemplos, el elemento de sintaxis puede codificarse en uno o más de un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) y un encabezado del segmento al que hace referencia el bloque actual de datos de vídeo. En algunos ejemplos, el elemento de sintaxis puede codificarse en el nivel de unidad de codificación (CU) de una CU que incluye el bloque actual de datos de vídeo.

Los requisitos de procesamiento y/o señalización del modelo de cuatro parámetros pueden ser inferiores a los requisitos de procesamiento y/o señalización del modelo de seis parámetros. Sin embargo, en algunos ejemplos, el modelo de seis parámetros puede dar como resultado bloques de predicción que coincidan mejor con el bloque que se está codificando, lo que puede reducir el tamaño de los valores residuales. Como tal, en algunos ejemplos, un codificador de vídeo puede equilibrar los costos de procesamiento y señalización de codificar un bloque mediante el uso de un modelo de seis parámetros contra los beneficios de valores residuales reducidos para el bloque y puede

seleccionar qué modelo es más ventajoso. De esta manera, las técnicas de esta divulgación pueden mejorar aún más la eficiencia de la compresión de vídeo mediante el uso de modelos de movimiento afines.

La Figura 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación y decodificación de vídeo 10 que puede utilizar técnicas para realizar la compensación de movimiento afin de esta divulgación. Como se muestra en la Figura 1, el sistema 10 incluye un dispositivo fuente 12 que proporciona datos de vídeo codificados a decodificar en un momento posterior mediante un dispositivo de destino 14. En particular, el dispositivo fuente 12 proporciona los datos de vídeo al dispositivo de destino 14 a través de un medio legible por ordenador 16. El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, que incluyen ordenadores de escritorio, ordenadores portátiles (es decir, laptop), tabletas, módulos de conexión, teléfonos móviles tal como los denominados teléfonos "inteligentes", las denominadas tabletas "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de vídeo en directo o similares. En algunos casos, el dispositivo fuente 12 y el dispositivo de destino 14 se pueden equipar para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados para decodificarlos a través de un medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de vídeo codificados desde el dispositivo fuente 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para permitir que el dispositivo fuente 12 transmita datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o alámbrica, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red en base a paquetes, tal como una red de área local, una red de área amplia, o una red global tal como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base, o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo fuente 12 al dispositivo de destino 14.

En algunos ejemplos, los datos codificados pueden salir desde la interfaz de salida 22 hacia un dispositivo de almacenamiento. De manera similar, puede accederse a los datos codificados desde el dispositivo de almacenamiento mediante la interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede almacenar el vídeo codificado que se genera mediante el dispositivo fuente 12. El dispositivo de destino 14 puede acceder a los datos de vídeo que se almacenan desde el dispositivo de almacenamiento a través de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar los datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a la red (NAS), o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, que incluye una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (por ejemplo, DSL, módem por cable, etc.), o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de las mismas.

Las técnicas de esta divulgación no se limitan necesariamente a las aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como difusión de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en transmisión continua por Internet, tales como transmisión continua dinámica adaptativa a través de HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, decodificación de vídeo digital que se almacenan en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 se puede configurar para soportar la transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tal como la transmisión de vídeo en directo, la reproducción de vídeo, la difusión de vídeo, y/o la telefonía de vídeo.

En el ejemplo de la Figura 1, el dispositivo fuente 12 incluye la fuente de vídeo 18, el codificador de vídeo 20, y la interfaz de salida 22. El dispositivo de destino 14 incluye la interfaz de entrada 28, el decodificador de vídeo 30 y el dispositivo de visualización 32. De acuerdo con esta divulgación, el codificador de vídeo 20 del dispositivo fuente 12 puede configurarse para aplicar las técnicas para realizar la compensación de movimiento afin de esta divulgación. En otros ejemplos, un dispositivo fuente y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 12 puede recibir datos de vídeo de una fuente de vídeo externa 18, tal como una cámara externa. Asimismo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema 10 que se ilustra en la Figura 1 es simplemente un ejemplo. Las técnicas para realizar la compensación de movimiento afín de esta divulgación se pueden realizar mediante cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque generalmente las técnicas de esta divulgación se realizan mediante un dispositivo de codificación de vídeo, las técnicas también se pueden realizar mediante un codificador/decodificador de vídeo, típicamente denominado como "Códec". Además, las técnicas de esta divulgación también pueden realizarse mediante un preprocesador de vídeo. El dispositivo fuente 12 y el dispositivo de destino 14 son simplemente ejemplos de tales dispositivos de codificación en los que el dispositivo fuente 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, el dispositivo 12 y 14 pueden operar de una manera sustancialmente simétrica de manera que cada uno de los dispositivos 12 y 14 incluya componentes de codificación y decodificación de vídeo. De ahí que, el sistema 10 puede soportar la transmisión de vídeo unidireccional o bidireccional entre dispositivos de vídeo 12, 14, por ejemplo, para transmisión de vídeo en directo, reproducción de vídeo, difusión de vídeo o videotelefonía.

La fuente de vídeo 18 del dispositivo fuente 12 puede incluir un dispositivo de captura de vídeo, tales como una cámara de vídeo, un archivo de vídeo que contiene vídeo que se captura previamente y/o una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo. Como una alternativa adicional, la fuente de vídeo 18 puede generar datos basados en gráficos de ordenador como la fuente de vídeo, o una combinación de vídeo en vivo, vídeo que se archiva y vídeo que se genera por ordenador. En algunos casos, si la fuente de vídeo 18 es una cámara de vídeo, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o videoteléfonos. Sin embargo, como se mencionó anteriormente, las técnicas descritas en esta divulgación pueden aplicarse a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o alámbricas. En cada caso, el vídeo que se captura, se precaptura o se genera por ordenador puede codificarse mediante el codificador de vídeo 20. La información de vídeo codificada puede luego salir mediante la interfaz de salida 22 a un medio legible por ordenador 16.

El medio legible por ordenador 16 puede incluir medios transitorios, tales como una difusión inalámbrica o una transmisión de red por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como un disco duro, una unidad flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otro medio legible por ordenador. En algunos ejemplos, un servidor de red (que no se muestra) puede recibir datos de vídeo codificados desde el dispositivo fuente 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, a través de la transmisión de red. De manera similar, un dispositivo informático de una instalación de producción de medios, tal como una instalación de estampado de discos, puede recibir datos de vídeo codificados desde el dispositivo fuente 12 y producir un disco que contenga los datos de vídeo codificados. Por lo tanto, puede entenderse que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de varias formas, en varios ejemplos.

La interfaz de entrada 28 del dispositivo de destino 14 recibe información del medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información de sintaxis que se define mediante el codificador de vídeo 20, que se usa además, por el decodificador de vídeo 30, que incluye elementos de sintaxis que describen características y/o procesamiento de bloques y otras unidades codificadas. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico emisor de luz (OLED), u otro tipo de dispositivo de visualización.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden operar de acuerdo con un estándar de codificación de vídeo, tal como el estándar de codificación de vídeo de alta eficiencia (HEVC) también denominado como ITU-T H.265. De manera alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden operar de acuerdo con otros estándares de propiedad o de la industria, tal como el estándar ITU-T H.264, alternativamente denominado como MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), o con las extensiones de tales estándares. Las técnicas de esta divulgación, sin embargo, no se limitan a ningún estándar de codificación particular. Otros ejemplos de estándares de codificación de vídeo incluyen MPEG-2 e ITU-T H.263. Aunque no se muestra en la Figura 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 se pueden integrar cada uno con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si aplica, las unidades MUX-DEMUX pueden adecuarse al protocolo multiplexor H.223 de la ITU, o a otros protocolos tales como el protocolo de datagramas de usuario (UDP).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, circuitos de procesamiento (incluidos los circuitos de función fija y/o circuitos de procesamiento programables), procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matriz de puertas programables en campo (FPGA), lógica discreta, software, hardware, microprograma o cualquier combinación de los mismos. Cuando las técnicas se implementan de manera parcial en el software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador no transitorio adecuado y ejecutar las instrucciones en el hardware mediante el uso de uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno del codificador de vídeo 20 y el decodificador de vídeo 30 puede incluirse en uno o más codificadores o decodificadores, cualquiera

de los cuales puede integrarse como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo.

En general, de acuerdo con ITU-T H.265, una imagen de vídeo se puede dividir en una secuencia de unidades de árbol de codificación (CTU) (o unidades de codificación más grandes (LCU)) que pueden incluir muestras tanto de luma como de croma. Alternativamente, las CTU pueden incluir datos monocromáticos (es decir, sólo muestras de luma). Los datos de sintaxis dentro de un flujo de bits pueden definir un tamaño para la CTU, que es una unidad de codificación más grande en términos de número de píxeles. Un segmento incluye un número de las CTU consecutivos en orden de codificación. Una imagen de vídeo se puede dividir en uno o más segmentos. Cada CTU se puede dividir en unidades de codificación (CU) de acuerdo con un árbol cuádruple. En general, una estructura de datos de árbol cuádruple incluye un nodo por CU, donde un nodo raíz corresponde a la CTU. Si una CU se divide en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU.

Cada nodo de la estructura de datos en árbol cuádruple puede proporcionar datos de sintaxis para la correspondiente CU. Por ejemplo, un nodo en el árbol cuádruple puede incluir una bandera de división, que indica si la CU correspondiente al nodo se divide en sub-CU. Los elementos de sintaxis para una CU pueden definirse recursivamente y pueden depender de si la CU se divide en sub-CU. Si una CU no se divide más, se refiere a ella como una CU hoja. En esta divulgación, cuatro sub-CU de una CU de hoja también se denominarán como CU hoja incluso si no hay una división explícita de la CU hoja original. Por ejemplo, si una CU de tamaño 16x16 no se divide más, las cuatro sub-CU de 8x8 también se denominarán como CU hoja, aunque la CU de 16x16 nunca se dividió.

En general, una CU tiene un propósito similar a un macrobloque del estándar H.264, excepto que una CU no tiene distinción de tamaño. Por ejemplo, una CTU puede dividirse en cuatro nodos hijos (también denominados sub-CU), y cada nodo hijo puede ser a su vez un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final no dividido, denominado nodo hoja del árbol cuádruple, comprende un nodo de codificación, también denominado como CU hoja. Los datos de sintaxis asociados con un flujo de bits codificado pueden definir un número máximo de veces que una CTU se puede dividir, denominado profundidad máxima de CU y también pueden definir un tamaño mínimo de los nodos de codificación. En consecuencia, un flujo de bits también puede definir una unidad de codificación más pequeña (SCU). Esta divulgación usa el término "bloque" para referirse a cualquiera de una CU, unidad de predicción (PU) o unidad de transformación (TU), en el contexto de HEVC, o estructuras de datos similares en el contexto de otros estándares (por ejemplo, macrobloques y subbloques del mismo en H.264/AVC).

Una CU incluye un nodo de codificación y las unidades de predicción (PU) y las unidades de transformación (TU) asociadas con el nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y es generalmente de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño de la CTU con un tamaño máximo, por ejemplo, de 64x64 píxeles o superior. Cada CU puede contener una o más PU y una o más TU. Los datos de sintaxis asociados con una CU pueden describir, por ejemplo, la partición de la CU en una o más PU. Los modos de partición pueden diferir entre si la CU es codificada en modo directo o en modo de salto, codificada en modo de intrapredicción, o codificada en modo de interpredicción. Las PU se pueden particionar para que sean de forma no cuadrada. Los datos de sintaxis asociados con una CU también pueden describir, por ejemplo, la partición de la CU en una o más TU de acuerdo con un árbol cuádruple. Una TU puede ser de forma cuadrada o no cuadrada (por ejemplo, rectangular).

El estándar HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU se basa típicamente en el tamaño de las PU (o particiones de una CU) dentro de una CU dada definida para una CTU particionada, aunque esto puede no ser siempre así. Las TU son típicamente del mismo tamaño o más pequeñas que las PU (o particiones de una CU, por ejemplo, en el caso de intrapredicción). En algunos ejemplos, las muestras residuales correspondientes a una CU se pueden subdividir en unidades más pequeñas mediante el uso de una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos hoja del RQT pueden denominarse como unidades de transformación (TU). Los valores de diferencia de píxeles asociados con las TU se pueden transformar para producir coeficientes de transformación, que se pueden cuantificar.

Una CU hoja puede incluir una o más unidades de predicción (PU) cuando se predice mediante el uso de la interpredicción. En general, una PU representa un área espacial correspondiente a toda o una porción de la CU correspondiente, y puede incluir datos para recuperar y/o generar una muestra de referencia para la PU. Además, una PU incluye datos relacionados con la predicción. Cuando la CU se codifica entre modos, una o más PU de la CU pueden incluir datos que definen información de movimiento, tal como uno o más vectores de movimiento, o las PU pueden codificarse en modo de salto. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento, y/o una lista de imágenes de referencia (por ejemplo, Lista 0, Lista 1) para el vector de movimiento.

Las CU hoja también se pueden predecir intramodo. En general, la intrapredicción implica predecir una CU hoja (o

particiones de la misma) mediante el uso de un intramodo. Un codificador de vídeo puede seleccionar un conjunto de píxeles vecinos, previamente codificados, para la CU hoja para usarlos para predecir la CU hoja (o particiones de la misma).

Una CU hoja también puede incluir una o más unidades de transformación (TU). Las unidades de transformación se pueden especificar mediante el uso de un RQT (también denominado como estructura de árbol cuádruple TU), como se divulgó anteriormente. Por ejemplo, una bandera de división puede indicar si una CU hoja está dividida en cuatro unidades de transformación. Luego, cada TU puede dividirse aún más en más sub-TU. Cuando una TU no se divide más, puede denominarse TU hoja. Generalmente, para la intracodificación, todas las TU hoja que pertenecen a una CU hoja comparten el mismo modo de intrapredicción. Es decir, generalmente se aplica el mismo modo de intrapredicción para calcular los valores predichos para todas las TU de una CU hoja. Para la intracodificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja mediante el uso del modo de intrapredicción, como una diferencia entre la porción de la CU correspondiente a la TU y el bloque original. Una TU no está necesariamente limitada al tamaño de una PU. Por tanto, las TU pueden ser más grandes o más pequeñas que una PU. Para la intracodificación, las particiones de una CU, o la propia CU, pueden ubicarse con una TU hoja correspondiente para la CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder al tamaño de la CU hoja correspondiente.

Además, las TU de las CU hoja también pueden asociarse con estructuras de datos de árbol cuádruple respectivas, denominadas árboles cuádruples residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuádruple que indica cómo se particiona la CU hoja en las TU. El nodo raíz de un árbol cuádruple TU generalmente corresponde a una CU hoja, mientras que el nodo raíz de un árbol cuádruple CU generalmente corresponde a una CTU (o LCU). Las TU del RQT que no están divididas se denominan como TU hoja. En general, esta divulgación usa los términos CU y TU para referirse a la CU hoja y la TU hoja, respectivamente, a menos que se indique de cualquier otra manera.

Una secuencia de vídeo típicamente incluye unos fotogramas o imágenes de vídeo, que comienzan con una imagen de punto de acceso aleatorio (RAP). Una secuencia de vídeo puede incluir datos de sintaxis en un conjunto de parámetros de secuencia (SPS) que caracterizan la secuencia de vídeo. Cada segmento de una imagen puede incluir los datos de sintaxis del segmento que describen un modo de codificación para el segmento respectivo. El codificador de vídeo típicamente opera en los bloques de vídeo dentro de los segmentos de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables, y pueden diferir en tamaño de acuerdo con un estándar de codificación específico.

Como ejemplo, se puede realizar predicción para las PU de varios tamaños. Si se supone que el tamaño de una CU particular es $2N \times 2N$, la intrapredicción se puede realizar en tamaños de PU de $2N \times 2N$ o $N \times N$, y la interpredicción se puede realizar en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. Las partición asimétrica para la interpredicción también puede realizarse para los tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$, y $nR \times 2N$. En la partición asimétrica, una dirección de una CU no se particiona, mientras que la otra dirección se particiona en 25 % y 75 %. La porción de la CU correspondiente a la partición del 25 % se indica mediante una "n" seguida de una indicación de "Arriba", "Abajo", "Izquierda", o "Derecha". Por tanto, por ejemplo, " $2N \times nU$ " se refiere a una CU de $2N \times 2N$ que se particiona de manera horizontal con una PU de $2N \times 0,5N$ en la parte superior y una PU de $2N \times 1,5N$ en la parte inferior.

En esta divulgación, " $N \times N$ " y " N por N " se pueden utilizar indistintamente para referirse a las dimensiones de los píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque $N \times N$ generalmente tiene N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, los bloques no necesitan tener necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .

El recuento del orden de las imágenes (POC) se usa ampliamente en los estándares de codificación de vídeo para identificar el orden de visualización de una imagen. Aunque hay casos en los que dos imágenes dentro de una secuencia de vídeo codificada pueden tener el mismo valor de POC, típicamente no ocurre dentro de una secuencia de vídeo codificada. Cuando hay múltiples secuencias de vídeo codificadas presentes en un flujo de bits, las imágenes con el mismo valor de POC pueden estar más cercanas entre sí en términos de orden de decodificación. Los valores de POC de las imágenes se usan típicamente para la construcción de listas de imágenes de referencia, la derivación del conjunto de imágenes de referencia como en HEVC y el escalado del vector de movimiento.

La compensación de movimiento en HEVC se usa para generar un predictor para el bloque inter actual. Se usa un vector de movimiento con precisión de un cuarto de píxel y los valores de píxeles en posiciones fraccionarias se interpolan mediante el uso de valores de píxeles enteros vecinos para los componentes tanto de luma como de croma.

En HEVC, para cada bloque, puede estar disponible un conjunto de información de movimiento. Un conjunto de información de movimiento contiene información de movimiento para direcciones de predicción hacia delante y hacia atrás. Aquí, las direcciones de predicción hacia delante y hacia atrás son dos direcciones de predicción de un modo de predicción bidireccional y los términos "hacia delante" y "hacia atrás" no tienen necesariamente un significado geométrico; en cambio, corresponden a la lista de imágenes de referencia 0 (RefPicList0) y a la lista de imágenes de referencia 1 (RefPicList1) de una imagen actual. Cuando solo se dispone de una lista de imágenes de referencia para una imagen o segmento, solo se dispone de RefPicList0 y la información de movimiento de cada bloque de un segmento siempre es hacia delante.

Para cada dirección de predicción, la información de movimiento debe contener un índice de referencia y un vector de movimiento. En algunos casos, por simplicidad, un vector de movimiento en sí mismo puede referirse de forma que se supone que tiene un índice de referencia asociado. Se usa un índice de referencia para identificar una imagen de referencia en la lista de imágenes de referencia actual (RefPicList0 o RefPicList1). Un vector de movimiento tiene una componente horizontal y una vertical.

En el estándar HEVC, hay dos modos de interpredicción, denominados modos de fusión (el salto se considera un caso especial de fusión) y modos de predicción de vector de movimiento avanzada (AMVP), respectivamente, para una unidad de predicción (PU). Ya sea en modo de AMVP o de fusión, se mantiene una lista de candidatos de vector de movimiento (MV) para múltiples predictores de vectores de movimiento. El(Los) vector(es) de movimiento, así como también los índices de referencia en el modo de fusión, de la PU actual se generan al tomar un candidato de la lista de candidatos de MV.

La lista de candidatos de MV contiene hasta cinco candidatos para el modo de fusión y sólo dos candidatos para el modo de AMVP. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento correspondientes tanto a las listas de imágenes de referencia (lista 0 y lista 1) como a los índices de referencia. Si un candidato de fusión se identifica mediante un índice de fusión, se usan las imágenes de referencia para la predicción de los bloques actuales, así como también se determinan los vectores de movimiento asociados. Sin embargo, en el modo de AMVP para cada dirección de predicción potencial o bien de la lista 0 o la lista 1, es necesario señalar explícitamente un índice de referencia, junto con un índice de MVP para la lista de candidatos de MV, ya que el candidato de AMVP contiene sólo un vector de movimiento. En el modo de AMVP, los vectores de movimiento predichos se pueden refinar aún más.

Como se puede ver arriba, un candidato de fusión puede corresponder a un conjunto completo de información de movimiento, mientras que un candidato de AMVP puede contener solo un vector de movimiento para una dirección de predicción y un índice de referencia específicos. Los candidatos para ambos modos se derivan de manera similar de los mismos bloques vecinos espaciales y temporales. Más abajo se divulgan más detalles de los candidatos vecinos espaciales para los modos de fusión y AMVP con referencia a la Figura 4.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden configurarse para realizar compensación de movimiento mediante el uso de modelos de movimiento afines. Por ejemplo, en lugar de usar únicamente un modelo de movimiento traslacional con un único vector de movimiento bidimensional (es decir, como en HEVC), el codificador de vídeo 20 y el decodificador de vídeo 30 pueden utilizar un modelo de movimiento afín que incluya múltiples vectores de movimiento. Más abajo se divulgan más detalles del uso de modelos de movimiento afines.

Después de la codificación intrapredictiva o interpredictiva mediante el uso de las PU de una CU, el codificador de vídeo 20 puede calcular los datos residuales para las TU de la CU. Las PU pueden comprender los datos de sintaxis que describen un procedimiento o modo de generación de datos de píxel predictivos en el dominio espacial (también denominado dominio de píxeles) y las TU pueden comprender los coeficientes en el dominio de transformación después la aplicación de una transformación, por ejemplo, una transformación de coseno discreta (DCT), una transformación de enteros, una transformación de ondícula, o una transformación conceptualmente similar a los datos de vídeo residuales. Los datos residuales pueden corresponder a las diferencias de píxeles entre los píxeles de la imagen no codificada y los valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU para incluir coeficientes de transformación cuantificados representativos de los datos residuales para la CU. Es decir, el codificador de vídeo 20 puede calcular los datos residuales (en forma de un bloque residual), transformar el bloque residual para producir un bloque de coeficientes de transformación y luego cuantificar los coeficientes de transformación para formar los coeficientes de transformación cuantificados. El codificador de vídeo 20 puede formar una TU que incluye los coeficientes de transformación cuantificados, así como también otra información de sintaxis (por ejemplo, información de división para la TU).

Como se indicó anteriormente, después de cualquier transformación para producir los coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación se refiere generalmente a un procedimiento en el que los coeficientes de transformación se cuantifican para de manera posible reducir la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El procedimiento de cuantificación puede reducir la profundidad de bits que se asocia con algunos o todos los coeficientes. Por ejemplo, un valor de n bits se puede redondear hacia abajo a un valor de m bits durante la cuantificación, donde n es mayor que m .

Después de la cuantificación, el codificador de vídeo puede escanear los coeficientes de transformación, lo que produce un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantificados. El escaneo puede diseñarse para colocar los coeficientes de mayor energía (y por lo tanto de menor frecuencia) en la parte frontal de la matriz y para colocar los coeficientes de menor energía (y por lo tanto de mayor frecuencia) en la parte posterior de la matriz. En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de escaneo predefinido para escanear los coeficientes de transformación cuantificados para producir un vector serializado que puede codificarse por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneo adaptativo. Después de escanear los coeficientes de transformación cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto en base a sintaxis (SBAC), la codificación de entropía de partición de intervalo de probabilidad (PIPE), u otra metodología de codificación de entropía. El codificador de vídeo 20 también puede codificar por entropía los elementos de sintaxis asociados con los datos de vídeo codificados para su uso por el decodificador de vídeo 30 al decodificar los datos de vídeo.

Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto puede relacionarse, por ejemplo, con si los valores vecinos del símbolo son de valor cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para transmitir un símbolo a transmitir. Las palabras código en VLC se pueden construir de manera que los códigos relativamente más cortos correspondan a los símbolos más probables, mientras que los códigos más largos correspondan a los símbolos menos probables. De esta manera, el uso de VLC puede lograr un ahorro de bits con respecto a, por ejemplo, el uso de palabras de código de igual longitud para cada símbolo a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.

En general, el decodificador de vídeo 30 realiza un procedimiento sustancialmente similar, aunque recíproco, al que se realiza por el codificador de vídeo 20 para decodificar los datos codificados. Por ejemplo, el decodificador de vídeo 30 cuantifica inversamente y transforma inversamente los coeficientes de una TU que se recibe para reproducir un bloque residual. El decodificador de vídeo 30 usa un modo de predicción señalizado (intra o interpredicción) para formar un bloque predicho. Luego, el decodificador de vídeo 30 combina el bloque predicho y el bloque residual (sobre una base píxel a píxel) para reproducir el bloque original. El procesamiento adicional puede realizarse, tal como realizar un procedimiento de desbloqueo para reducir los artefactos visuales a lo largo de los límites del bloque. Además, el decodificador de vídeo 30 puede decodificar elementos de sintaxis mediante el uso de la CABAC de una manera sustancialmente similar, aunque recíproca, al procedimiento de codificación de la CABAC del codificador de vídeo 20.

El codificador de vídeo 20 puede enviar además datos de sintaxis, tales como los datos de sintaxis que se basan en bloques, datos de sintaxis que se basan en imágenes y datos de sintaxis que se basan en secuencias, para el decodificador de vídeo 30, por ejemplo, en un encabezado de imagen, un encabezado de bloque, un encabezado del segmento, u otros datos de sintaxis, tales como un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un conjunto de parámetros de vídeo (VPS).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores o decodificadores adecuados, según proceda, tales como uno o más microprocesadores, circuitos de procesamiento (incluidos los circuitos de función fija y/o circuitos de procesamiento programables), procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matriz de puertas programables en campo (FPGA), circuitos lógicos discretos, software, hardware, microprograma o cualquier combinación de los mismos. Cada uno del codificador de vídeo 20 y decodificador de vídeo 30 se puede incluir en uno o más codificadores o decodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/decodificador combinado (CODEC). Un dispositivo que incluye el codificador de vídeo 20 y/o el decodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador, y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

La Figura 2 es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 20 que puede implementar técnicas para realizar la compensación de movimiento afín de esta divulgación. El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de los bloques de vídeo dentro de los segmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de un fotograma o imagen de vídeo determinado. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de fotogramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) puede referirse a cualquiera de varios modos de codificación que se basan en el espacio. Los intermodos, tales como la predicción unidireccional (modo P) o bipredicción (modo B), pueden referirse a cualquiera de varios modos de codificación que se basan en el tiempo.

Como se muestra en la Figura 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de un fotograma de vídeo a codificar. En el ejemplo de la Figura 2, el codificador de vídeo 20 incluye la unidad de selección de modo 40, la memoria de imágenes de referencia 64 (que también puede denominarse memoria intermedia de imagen decodificada (DPB)), un sumador 50, la unidad de procesamiento de transformación 52, la unidad de cuantificación

54 y la unidad de codificación de entropía 56. La unidad de selección de modo 40, a su vez, incluye la unidad de compensación de movimiento 44, la unidad de estimación de movimiento 42, la unidad de intrapredicción 46 y la unidad de partición 48. Para la reconstrucción del bloque de vídeo, el codificador de vídeo 20 también incluye la unidad de cuantificación inversa 58, la unidad de transformación inversa 60 y el sumador 62. También puede incluirse un filtro de desbloqueo (no se muestra en la Figura 2) para filtrar los límites de bloque para eliminar los artefactos de bloqueo del vídeo reconstruido. Si se desea, el filtro de desbloqueo típicamente filtraría la salida del sumador 62. Pueden además, usarse filtros adicionales (en bucle o bucle posterior) además del filtro de desbloqueo. Tales filtros no se muestran por brevedad, pero si se desea, pueden filtrar la salida del sumador 50 (como un filtro en bucle).

Durante el procedimiento de codificación, el codificador de vídeo 20 recibe un fotograma o segmento de vídeo a codificar. El fotograma o segmento puede dividirse en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan la codificación interpredictiva del bloque de vídeo que se recibe con relación a uno o más bloques en uno o más fotogramas de referencia para proporcionar una predicción temporal. La unidad de intrapredicción 46 puede realizar alternativamente una codificación intrapredictiva del bloque de vídeo que se recibe con relación a uno o más bloques vecinos en el mismo fotograma o segmento como el bloque a codificar para proporcionar predicción espacial. El codificador de vídeo 20 puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.

Además, la unidad de partición 48 puede particionar bloques de datos de vídeo en subbloques, en base a la evaluación de esquemas de partición previos en pasadas de codificación previas. Por ejemplo, la unidad de partición 48 puede particionar inicialmente un fotograma o segmento en las LCU y particionar cada una de las LCU en sub-CU en base al análisis de la tasa de distorsión (por ejemplo, optimización de la tasa de distorsión). La unidad de selección de modo 40 puede producir además, una estructura de datos de árboles cuádruples indicativa de la partición de una CTU en sub-CU. Las CU de los nodo hoja del árbol cuádruple pueden incluir una o más PU y una o más TU.

La unidad de selección de modo 40 puede seleccionar uno de los modos de predicción, intra o inter, por ejemplo, en base a los resultados de error, y proporciona el bloque predicho resultante al sumador 50 para generar datos residuales y al sumador 62 para reconstruir el bloque codificado para su uso como fotograma de referencia. La unidad de selección de modo 40 proporciona además, elementos de sintaxis, tales como vectores de movimiento, indicadores intramodo, información de partición y otra información de sintaxis similar, a la unidad de codificación de entropía 56.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar muy integradas, pero se ilustran por separado para propósitos conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el procedimiento de generar los vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro del fotograma o imagen de vídeo actual con relación a un bloque predictivo dentro de un fotograma de referencia (u otra unidad codificada) con relación al bloque actual que se está codificando dentro del fotograma actual (u otra unidad codificada). Un bloque predictivo es un bloque que coincide estrechamente con el bloque a codificar, en términos de diferencia de píxeles, que puede determinarse mediante la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular los valores para las posiciones de píxeles subenteros de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxeles fraccionarios de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con relación a las posiciones de píxeles completos y a las posiciones de píxeles fraccionarios y generar un vector de movimiento con precisión de píxeles fraccionarios.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un segmento intercodificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar de una primera lista de imágenes de referencia (List 0) o de una segunda lista de imágenes de referencia (List 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación de entropía 56 y a la unidad de compensación de movimiento 44.

La compensación de movimiento, que se realiza mediante la unidad de compensación de movimiento 44, puede implicar buscar o generar el bloque predictivo en base al vector de movimiento que se determina mediante la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente, en algunos ejemplos. Al recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual mediante la sustracción de los valores de píxeles del bloque predictivo, de los valores de píxeles

del bloque de vídeo actual que se codifica, lo que forma valores de diferencia de píxeles, como se divulga más abajo. En general, la unidad de estimación de movimiento 42 realiza una estimación de movimiento con relación a los componentes de luma, y la unidad de compensación de movimiento 44 usa vectores de movimiento calculados en base a los componentes de luma para ambos componentes de croma y componentes de luma. La unidad de selección de modo 40 puede además, generar elementos de sintaxis que se asocian con los bloques de vídeo y el segmento de vídeo para su uso mediante el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del segmento de vídeo.

El codificador de vídeo 20 puede configurarse para realizar cualquiera de las diversas técnicas de esta divulgación divulgadas anteriormente con respecto a la Figura 1, y como se describirá en más detalle más abajo. Por ejemplo, la unidad de compensación de movimiento 44 puede configurarse para codificar información de movimiento para un bloque de datos de vídeo mediante el uso de AMVP o modo de fusión de acuerdo con HEVC, y/o puede configurarse para codificar información de movimiento afín o un bloque de datos de vídeo mediante el uso del modo inter afín o modo de fusión afín de acuerdo con las técnicas de esta divulgación.

La unidad de intrapredicción 46 puede intrapredecir un bloque actual, como una alternativa a la interpredicción que se realiza mediante la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se describió anteriormente. En particular, la unidad de intrapredicción 46 puede determinar un modo de intrapredicción para usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 46 puede codificar un bloque actual mediante el uso de varios modos de intrapredicción, por ejemplo, durante pasadas de codificación separadas, y la unidad de intrapredicción 46 (o la unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado para usar de los modos probados.

Por ejemplo, la unidad de intrapredicción 46 puede calcular valores de tasa de distorsión mediante el uso de un análisis de la tasa de distorsión para los varios modos de intrapredicción probados, y seleccionar el modo de intrapredicción que tiene las mejores características de tasa de distorsión entre los modos probados. El análisis de la tasa de distorsión generalmente determina una cantidad de distorsión (o error) entre un bloque codificado y uno original, el bloque no codificado que se codificó para producir el bloque codificado, así como también una tasa de bits (es decir, un número de bits) usado para producir el bloque codificado. La unidad de intrapredicción 46 puede calcular las relaciones de las distorsiones y tasas para los varios bloques codificados para determinar qué modo de intrapredicción exhibe el mejor valor de tasa de distorsión para el bloque.

Después de seleccionar un modo de intrapredicción para un bloque, la unidad de intrapredicción 46 puede proporcionar información indicativa del modo de intrapredicción que se selecciona para el bloque a la unidad de codificación de entropía 56. La unidad de codificación de entropía 56 puede codificar la información que indica el modo de intrapredicción que se selecciona. El codificador de vídeo 20 puede incluir en los datos de configuración del flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índice en modo de intrapredicción y una pluralidad de tablas de índice en modo de intrapredicción modificadas (que se denominan además, tablas de asignación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índice de modo de intrapredicción y una tabla de índice de modo de intrapredicción modificada para usar en cada uno de los contextos.

El codificador de vídeo 20 forma un bloque de vídeo residual mediante la sustracción de los datos de predicción de la unidad de selección de modo 40 del bloque de vídeo original que se está codificando. El sumador 50 representa el componente o componentes que realizan esta operación de sustracción. La unidad de procesamiento de transformación 52 aplica una transformación, tal como una transformación de coseno discreta (DCT) o una transformación conceptualmente similar, al bloque residual, lo que produce un bloque de vídeo que comprende valores de coeficiente de transformación. En lugar de una DCT podrían usarse transformaciones de ondícula, transformaciones de enteros, transformaciones de subbanda, transformaciones de seno discreto (DST) u otros tipos de transformaciones. En cualquier caso, la unidad de procesamiento de transformación 52 aplica la transformación al bloque residual, lo que produce un bloque de coeficientes de transformación. La transformación puede convertir la información residual de un dominio de píxeles a un dominio de transformación, tal como un dominio de frecuencia. La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformación para reducir aún más la tasa de bits. El procedimiento de cuantificación puede reducir la profundidad de bits que se asocia con algunos o todos los coeficientes. El grado de cuantificación puede modificarse mediante el ajuste de un parámetro de cuantificación.

Después de la cuantificación, la entropía de la unidad de codificación de entropía 56 codifica los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación de entropía 56 puede realizar codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto en base a sintaxis (SBAC), codificación de entropía de partición de intervalo de probabilidad (PIPE) u otra técnica de codificación de entropía. En el caso de la codificación de entropía basada en el contexto, el contexto puede basarse en bloques vecinos. Después de la codificación de entropía mediante la unidad de codificación de entropía 56, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, decodificador de vídeo 30) o archivarse para su posterior transmisión o recuperación.

La unidad de cuantificación inversa 58 y la unidad de transformación inversa 60 aplican la cuantificación inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio del píxel. En particular, el sumador 62 adiciona el bloque residual reconstruido al bloque de predicción de compensación de movimiento producido anteriormente por la unidad de compensación de movimiento 44 o la unidad de intrapredicción 46 para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede usarse mediante la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para intercodificar un bloque en un fotograma de vídeo subsiguiente.

La Figura 3 es un diagrama de bloques que ilustra un ejemplo del decodificador de vídeo 30 que puede implementar técnicas para realizar la compensación de movimiento afín de esta divulgación. En el ejemplo de la Figura 3, el decodificador de vídeo 30 incluye una unidad de decodificación de entropía 70, una unidad de compensación de movimiento 72, una unidad de intrapredicción 74, una unidad de cuantificación inversa 76, una unidad de transformación inversa 78, una memoria de imágenes de referencia 82 y un sumador 80. El decodificador de vídeo 30 puede, en algunos ejemplos, realizar una pasada de decodificación generalmente recíproca a la pasada de codificación que se escribe con respecto al codificador de vídeo 20 (Figura 2). La unidad de compensación de movimiento 72 puede generar datos de predicción en base a los vectores de movimiento que se reciben desde la unidad de decodificación de entropía 70, mientras que la unidad de intrapredicción 74 puede generar datos de predicción en base a los indicadores de modo de intrapredicción que se reciben desde la unidad de decodificación de entropía 70.

Durante el procedimiento de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa los bloques de vídeo de un segmento de vídeo codificado y los elementos de sintaxis asociados del codificador de vídeo 20. La unidad de decodificación de entropía 70 del decodificador de vídeo 30 decodifica la entropía del flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción, y otros elementos de sintaxis. La unidad de decodificación de entropía 70 envía los vectores de movimiento y otros elementos de sintaxis a la unidad de compensación de movimiento 72. El decodificador de vídeo 30 puede recibir los elementos de sintaxis a nivel de segmento de vídeo y/o a nivel de bloque de vídeo.

Cuando el segmento de vídeo se codifica como un segmento intracodificado (I), la unidad de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del segmento de vídeo actual en base a un modo de intrapredicción señalado y datos de bloques previamente decodificados del fotograma o imagen actual. Cuando la trama de vídeo se codifica como un segmento intercodificado (es decir, B o P), la unidad de compensación de movimiento 72 produce bloques predictivos para un bloque de vídeo del segmento de vídeo actual en base a los vectores de movimiento y otros elementos de sintaxis que se reciben de la unidad de decodificación de entropía 70. Los bloques predictivos se pueden producir a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de fotogramas de referencia, List 0 y List 1, mediante el uso de las técnicas de construcción por defecto en base a las imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del segmento de vídeo actual al analizar los vectores de movimiento y otros elementos de sintaxis, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos de sintaxis que se reciben para determinar un modo de predicción (por ejemplo, intra o interpredicción) que se usa para codificar los bloques de vídeo del segmento de vídeo, un tipo de segmento de interpredicción (por ejemplo, el segmento B o el segmento P), la información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de vídeo intercodificado del segmento, el estado de interpredicción para cada bloque de vídeo intercodificado del segmento, y otra información para decodificar los bloques de vídeo en el segmento de vídeo actual.

El decodificador de vídeo 30 puede configurarse para realizar cualquiera de las diversas técnicas de esta divulgación divulgadas anteriormente con respecto a la Figura 1, y como se divulgará en más detalle más abajo. Por ejemplo, la unidad de compensación de movimiento 72 puede configurarse para realizar predicción de vector de movimiento mediante el uso de AMVP o modo de fusión de acuerdo con HEVC, y/o puede configurarse para realizar la información de movimiento afín o un bloque de datos de vídeo mediante el uso del modo inter afín o modo de fusión afín de acuerdo con las técnicas de esta divulgación. La unidad de decodificación de entropía 70 puede decodificar uno o más elementos de sintaxis que representan cómo se codifica la información de movimiento para el bloque actual.

La unidad de compensación de movimiento 72 puede además, realizar una interpolación en base a filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como se usan mediante el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular valores interpolados para píxeles subenteros de bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación que se usan mediante el codificador de vídeo 20 a partir de los elementos de sintaxis que se reciben y usar los filtros de interpolación para producir bloques predictivos.

La unidad de cuantificación inversa 76 cuantifica inversamente, es decir, descuantifica, los coeficientes de transformación cuantificados que se proporcionan en el flujo de bits y decodificados mediante la unidad de decodificación de entropía 70. El procedimiento de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QP_Y que se calcula mediante el decodificador de vídeo 30 para cada bloque de vídeo en el segmento de vídeo para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse.

La unidad de transformación inversa 78 aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación de entero inversa o un procedimiento de transformación inversa conceptualmente similar, a los coeficientes de transformación con el fin de producir bloques residuales en el dominio del píxel.

Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual en base a los vectores de movimiento y otros elementos de sintaxis, el decodificador de vídeo 30 forma un bloque de vídeo decodificado mediante la suma de los bloques residuales de la unidad de transformación inversa 78 con los correspondientes bloques predictivos que se generan mediante la unidad de compensación de movimiento 72. El sumador 80 representa el componente o componentes que realizan esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques decodificados con el fin de eliminar los artefactos de bloqueo. Pueden además, usarse otros filtros en bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de cualquier otra manera la calidad del vídeo. Los bloques de vídeo decodificados en un fotograma o imagen dada se almacenan luego en la memoria de imágenes de referencia 82, que almacena las imágenes de referencia usadas para la compensación de movimiento subsiguiente. La memoria de imágenes de referencia 82 también almacena el vídeo decodificado para una presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la Figura 1.

Las Figuras 4A y 4B son diagramas conceptuales que ilustran los candidatos a vecinos espaciales en codificación de vídeo de alta eficiencia (HEVC). Como se divulgó anteriormente, los candidatos de MV espaciales pueden derivarse de los bloques vecinos para una PU específica (PU_0), aunque los procedimientos que generan los candidatos a partir de los bloques difieren para los modos de fusión y AMVP.

La Figura 4A ilustra un ejemplo de cómo un codificador de vídeo puede derivar candidatos de MV espaciales en modo de fusión. En el modo de fusión, se pueden derivar hasta cuatro candidatos de MV espaciales con los órdenes que se muestran en la Figura 4A con números, y el orden es el siguiente: izquierda (0), superior (1), superior derecho (2), inferior izquierdo (3), y superior izquierdo (4), como se muestra en la Figura 4A.

La Figura 4B ilustra un ejemplo de cómo un codificador de vídeo puede derivar candidatos de MV espaciales en modo de AMVP. En el modo de AMVP, los bloques vecinos se dividen en dos grupos: el grupo de la izquierda que consta de los bloques 0 y 1, y el grupo superior que consta de los bloques 2, 3 y 4, como se muestra en la Figura 4B. Para cada grupo, el candidato potencial de un bloque vecino que hace referencia a la misma imagen de referencia que la indicada por el índice de referencia señalado tiene la máxima prioridad para ser elegido para formar un candidato final del grupo. Es posible que todos los bloques vecinos no contengan un vector de movimiento que apunte a la misma imagen de referencia. Por lo tanto, si no se puede encontrar tal candidato, el primer candidato disponible se escalará para formar el candidato final; por tanto se pueden compensar las diferencias de distancia temporal.

La Figura 5 es un diagrama conceptual que ilustra un vector de movimiento afin de dos puntos con cuatro parámetros afines. Como se muestra en la Figura 5, (v_{0x}, v_{0y}) denotado como v_0 es el CPMV para la esquina superior izquierda 502 del bloque actual 500 y (v_{1x}, v_{1y}) denotado como v_1 es el CPMV para la esquina superior derecha 504 del bloque actual 500. Como se divulgó anteriormente, los CPMV para el bloque actual 500 pueden formar un campo de vector de movimiento (MVF) que se representa de acuerdo con la Ecuación (2) anterior.

En el modelo de prueba JEM, la predicción de movimiento afin solo se aplica a bloques cuadrados. Como extensión natural, la predicción de movimiento afin se puede aplicar a bloques no cuadrados.

La Figura 6 es un diagrama conceptual que ilustra un modo inter afin. Para bloques (por ejemplo, las CU/PU) que tienen un tamaño igual o mayor que 16×16 , un codificador de vídeo (por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30) puede aplicar un modo inter afin (AF_INTER) como sigue. En algunos ejemplos, si el bloque actual (por ejemplo, la CU/PU actual) está en modo inter afin, el codificador de vídeo puede señalar una bandera afin en el nivel de la CU/PU en el flujo de bits. El codificador de vídeo puede construir una lista de vectores de movimiento candidatos para el actual mediante el uso de vectores de movimiento de bloques vecinos válidos reconstruidos del bloque actual. Por ejemplo, como se muestra en el ejemplo de la Figura 6, los predictores de vectores de movimiento candidatos para el CPMV superior izquierdo v_0 pueden ser seleccionados de los vectores de movimiento del bloque 602A, 602B y 602C (es decir, bloques vecinos en contacto con la esquina superior izquierda del bloque actual 600). El codificador de vídeo puede escalar el vector de movimiento del bloque vecino de acuerdo con la lista de referencias y la relación entre el POC de la referencia para el bloque vecino, el POC de la referencia para la CU/PU actual y el POC de la CU/PU actual. El codificador de vídeo puede realizar un enfoque similar para seleccionar predictores de vectores de movimiento candidatos para el CPMV superior derecho v_1 desde el bloque

vecino 602D y 602E (es decir, bloques vecinos en contacto con la esquina superior derecha del bloque actual 600). Como tal, en algunos ejemplos, la lista de candidatos se puede representar como $\{(v_0, v_1) | v_0 = \{v_{602A}, v_{602B}, v_{602C}\}, v_1 = \{v_{602D}, v_{602E}\}\}$.

Si el número de la lista de candidatos es menor que un umbral (por ejemplo, dos, tres o cuatro), el codificador de vídeo puede asignar los candidatos de AMVP a v_0 y v_1 . El codificador de vídeo puede utilizar el costo de optimización de la tasa de distorsión (RDO) del bloque actual para determinar que (v_0, v_1) seleccionar como predicción del vector de movimiento del punto de control (CPMVP) del bloque actual. El codificador de vídeo puede señalar el índice para indicar la posición del CPMVP en la lista de candidatos en el flujo de bits.

En base al CPMVP del bloque afín actual, el codificador de vídeo puede aplicar una estimación de movimiento afín para determinar el CPMV. El codificador de vídeo puede codificar una representación de una diferencia entre el CPMV y el CPMVP en el flujo de bits.

El codificador de vídeo puede realizar la predicción de compensación de movimiento afín como se describió anteriormente para generar los residuos del bloque actual. El codificador de vídeo puede transformar y cuantificar los residuos que se generan del bloque actual y codificar los residuos cuantificados en el flujo de bits (por ejemplo, de una manera similar a HEVC).

Las Figuras 7A y 7B son diagramas conceptuales que ilustran candidatos para un modo de fusión afín. Cuando se aplica el modo de fusión afín (AF_MERGE) a un bloque actual, un codificador de vídeo (por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30) puede obtener el primer bloque codificado con el modo afín a partir de los bloques vecinos válidos reconstruidos del bloque actual. En algunos ejemplos, el codificador de vídeo puede analizar los bloques vecinos reconstruidos en un orden de selección particular para obtener el primer bloque codificado con el modo afín. La Figura 7A ilustra un ejemplo de orden de selección. Como se muestra en la Figura 7A, el orden de selección puede ser el siguiente: bloque izquierdo 702A, bloque superior 702B, bloque superior derecho 702C, bloque inferior izquierdo 702D, hasta el bloque superior izquierdo 702E.

La Figura 7B ilustra un ejemplo en el que el bloque izquierdo es el primer bloque en el orden de selección codificado con el más afín. Como se muestra en la Figura 7B, el codificador de vídeo puede derivar los vectores de movimiento de la esquina superior izquierda (v_2), la esquina superior derecha (v_s) y la esquina inferior izquierda (v_4) de la CU/PU 704 que contiene el bloque seleccionado 1002A. El codificador de vídeo puede determinar/calcular el vector de movimiento de la esquina superior izquierda del bloque actual 700 (es decir, v_0) y el vector de movimiento de la esquina superior derecha del bloque actual 700 (es decir, v_1) en base a los vectores de movimiento derivados del bloque seleccionado (es decir, v_2, v_3 , y v_4).

El codificador de vídeo puede determinar el MVF del bloque actual 700 en base a los CPMV del bloque actual 700 v_0 y v_1 de acuerdo con el modelo de movimiento afín simplificado descrito anteriormente en la Ecuación (2). El codificador de vídeo puede aplicar la MCP afín mediante el uso del MVF como se describió anteriormente.

Para identificar si el bloque actual se codifica con el modo de fusión afín, el codificador de vídeo puede señalar una bandera afín en el flujo de bits cuando hay al menos un bloque vecino codificado en el modo afín. Si no existe ningún vecino de bloque afín para el bloque actual, el codificador de vídeo puede omitir la codificación de la bandera afín en el flujo de bits o puede codificar la bandera afín para indicar que no existe ningún vecino de bloque afín para el bloque actual.

Como se divulgó anteriormente, los procedimientos de modelo de movimiento afín existentes (por ejemplo, en el modelo de prueba JEM y la Contribución 1016) presentan varios problemas y/o tienen varias desventajas. Como ejemplo, en la Contribución 1016, el movimiento afín de cuatro parámetros ha planteado una restricción en los parámetros afines en el MVx y el MVy, lo que los fuerza a tener propiedades de escala simétrica. Esta restricción puede no ser cierta en el contenido de vídeo diversificado.

Como otro ejemplo, el modo de fusión afín se basa en un orden de verificación predefinido que se basa principalmente en la esquina inferior izquierda y la esquina superior derecha. Este orden predefinido ha colocado la esquina superior izquierda en la prioridad más baja, mientras que esta información de la esquina se usa mucho en la siguiente derivación del modelo afín.

Como otro ejemplo, la fusión afín solo puede heredar el modelo vecino al deformar la esquina del bloque vecino de MV a la esquina del bloque actual. No hay flexibilidad para cambiar o ajustar los parámetros del modelo afín cuando se hereda el modelo afín vecino.

De acuerdo con una o más técnicas de esta divulgación, un codificador de vídeo puede codificar un elemento de sintaxis que indica cómo se identifica un bloque predictor de datos de vídeo. Por ejemplo, un codificador de vídeo puede codificar un elemento de sintaxis que indica si se usa un modelo afín de cuatro parámetros o de seis parámetros para identificar un bloque predictor de datos de vídeo para un bloque actual de datos de vídeo. Al permitir la selección entre un modelo afín de cuatro parámetros y uno de seis parámetros, las técnicas de esta

divulgación pueden permitir que los vectores de movimiento tengan propiedades de escalado no simétrico, lo que puede mejorar la eficiencia de la codificación.

5 En algunos ejemplos, el codificador de vídeo puede codificar el elemento de sintaxis en el nivel de unidad de codificación (CU). Por ejemplo, se puede introducir una bandera en el nivel de CU para indicar si se usa un modelo de movimiento afín de cuatro parámetros o de seis parámetros para un bloque actual en la CU.

10 En algunos ejemplos, el codificador de vídeo puede codificar el elemento de sintaxis en una sintaxis en modo de salto o en una sintaxis en modo de fusión a la que hace referencia el bloque actual de datos de vídeo. Por ejemplo, se puede introducir una bandera en el modo de salto o de fusión para indicar si se usa un modelo de movimiento afín de cuatro parámetros o de seis parámetros para el bloque actual.

15 En algunos ejemplos, el codificador de vídeo puede codificar el elemento de sintaxis en una sintaxis intermodo a la que hace referencia el bloque actual de datos de vídeo. Por ejemplo, se puede introducir una bandera en el intermodo (si el bloque actual no está ni en modo de salto ni de fusión) para indicar si se usa un modelo de movimiento afín de cuatro parámetros o de seis parámetros para el bloque actual.

20 En algunos ejemplos, en lugar de indicar solo si un bloque predictor de datos de vídeo para un bloque actual de datos de vídeo se identifica mediante el uso de un modelo afín de cuatro parámetros o un modelo afín de seis parámetros, un codificador de vídeo puede codificar el elemento de sintaxis para indicar si un bloque predictor de datos de vídeo para un bloque actual de datos de vídeo se identifica mediante el uso de un único vector de movimiento, un modelo afín de cuatro parámetros, un modelo afín de seis parámetros o un modelo afín conmutable de cuatro/seis parámetros. Por ejemplo, un elemento de sintaxis en un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) y/o un encabezado del segmento puede estar presente para señalar

25 cuál de los siguientes casos se usa para la secuencia/imagen/segmento actual, 1) afín deshabilitado, 2) afín de 4 parámetros, 3) afín de 6 parámetros, 4) afín conmutable de 4/6. El elemento de sintaxis se puede codificar mediante el uso de una palabra de código unaria, unaria truncada o de longitud fija.

30 En realizaciones de la invención protegida, un codificador de vídeo codifica un elemento de sintaxis habilitante que indica si un número de parámetros usados en modelos afines usados para identificar bloques predictores de datos de vídeo es conmutable. Por ejemplo, el codificador de vídeo puede codificar una bandera en un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) y/o encabezado del segmento para indicar si el modelo afín conmutable está habilitado para imágenes que hacen referencia al SPS o al PPS o al encabezado del segmento. En las realizaciones de la invención protegida, la bandera se codifica en el SPS.

35 Cuando el elemento de sintaxis de habilitación indica que el número de parámetros usados en los modelos afines usados para identificar bloques predictores de datos de vídeo es conmutable (por ejemplo, cuando el elemento de sintaxis de habilitación es una bandera con valor 1), el codificador de vídeo, de acuerdo con la invención protegida, codifica un elemento de sintaxis que indica si se usa un modelo afín de cuatro parámetros o de seis parámetros para

40 identificar un bloque predictor de datos de vídeo para un bloque actual de datos de vídeo como se divulgó anteriormente. Por ejemplo, cuando el elemento de sintaxis de habilitación indica que el número de parámetros usados en modelos afines usados para identificar bloques predictores de datos de vídeo es conmutable (por ejemplo, cuando el elemento de sintaxis de habilitación es una bandera con valor 1), los modelos afines de cuatro y seis parámetros están ambos habilitados y se puede señalar una bandera adicional para cada bloque para indicar el

45 uso de modelos de cuatro o seis parámetros.

50 Cuando el elemento de sintaxis de habilitación indica que el número de parámetros usados en modelos afines usados para identificar bloques predictores de datos de vídeo no es conmutable (por ejemplo, cuando el elemento de sintaxis de habilitación es una bandera con valor 0), el codificador de vídeo puede determinar que se usa un modelo afín de cuatro parámetros (es decir, si se usa afín). En tales ejemplos, el codificador de vídeo puede omitir la codificación del elemento de sintaxis que indica si se usa un modelo afín de cuatro parámetros o de seis parámetros para identificar el bloque predictor de datos de vídeo para el bloque actual de datos de vídeo.

55 En algunos ejemplos, uno o más de los elementos de sintaxis descritos anteriormente (es decir, la bandera de parámetro afín (cuatro parámetros o seis parámetros) y/o el elemento de sintaxis de habilitación) pueden codificarse mediante el uso de un modelo de contexto de CABAC en función del uso de parámetros afines del bloque vecino. En un ejemplo, el índice de contexto del parámetro afín actual CtxVal depende de los bloques vecinos izquierdo y superior. Si el bloque vecino izquierdo no está disponible, o no está en modo afín, o afín de seis parámetros, el leftCtx se establece igual a 0; de cualquier otra manera (la izquierda disponible y el modo afín de seis parámetros) el

60 leftCtx se establece igual a 1. Se puede calcular un cálculo similar para el bloque vecino anterior para obtener el aboveCtx. Entonces, el CtxVal del bloque actual se establece igual a leftCtx+aboveCtx. En este caso, CtxVal está en el rango de [0, 2] inclusive. También son posibles otras variaciones de configuración del leftCtx (aboveCtx). Por ejemplo, el leftCtx (aboveCtx) se establece igual a 0 si el bloque vecino izquierdo (superior) no está disponible o no tiene codificación afín; 1 si el bloque vecino izquierdo (superior) usa afines de cuatro parámetros; 2 si el bloque vecino izquierdo (superior) usa afines de seis parámetros. En este caso, CtxVal está en el rango de [0, 4] inclusive.

65

En algunos ejemplos, uno o más de los elementos de sintaxis descritos anteriormente (es decir, la bandera de parámetro afín (de cuatro parámetros o seis parámetros) y/o el elemento de sintaxis de habilitación) pueden codificarse mediante el uso del modelo de contexto de CABAC en función del tamaño de bloque actual y puede usarse un umbral de tamaño de bloque para diferenciar los distintos contextos. Por ejemplo, el contexto 0 se usa para tamaños de bloque iguales o menores que 16x16; mientras que el contexto 1 se usa para tamaños de bloque mayores que 16x16. El umbral puede predefinirse o señalizarse en el flujo de bits. El tamaño del bloque podría especificarse mediante el ancho y la altura del bloque actual por separado o conjuntamente. Por ejemplo, el tamaño se puede representar mediante el valor de ancho*altura.

En algunos ejemplos, uno o más de los elementos de sintaxis descritos anteriormente (es decir, la bandera de parámetro afín (cuatro parámetros o seis parámetros) y/o el elemento de sintaxis de habilitación) también pueden codificarse mediante el uso de un modelo de derivación de CABAC sin ningún contexto.

La Figura 8 es un diagrama conceptual que ilustra un modelo de movimiento afín de seis parámetros, de acuerdo con una o más técnicas de esta divulgación. Un modelo afín de cuatro parámetros puede incluir dos vectores de movimiento y un modelo afín de seis parámetros puede incluir tres vectores de movimiento. En algunos ejemplos, tal como cuando se usa el modelo de movimiento afín de seis parámetros, un codificador de vídeo puede codificar tres diferencias de vector de movimiento (MVD) en el flujo de bits para el intermodo. Los tres predictores de vectores de movimiento pueden generarse a partir de vectores de movimiento vecinos o derivarse de los vectores de movimiento vecinos. Los vectores de movimiento vecinos pueden ser o no vectores de movimiento afines. Por ejemplo, tres vectores de movimiento en el bloque actual v_0 (MV0), v_1 (MV1), y v_2 (MV2) en las tres esquinas del bloque actual 800 pueden codificarse como se muestra en la Figura 8. Con el fin de predecir v_0 , los vectores de movimiento de 802A (superior izquierdo), 802B (superior) y 802C (izquierdo) son posibles candidatos. De manera similar, los vectores de movimiento de 802D (superior) y 802E (superior derecho) son posibles candidatos para predecir v_1 , y los vectores de movimiento de 802F (izquierda) y 802G (inferior izquierdo) son posibles candidatos para predecir v_2 . En algunos ejemplos, el primer candidato disponible para cada posición en un orden de verificación predefinido se usa directamente como su predictor.

Los tres predictores de vectores de movimiento se pueden seleccionar de una lista de combinaciones mediante el uso de un esquema de validación, clasificación y deduplicación y solo las primeras K combinaciones se usan como posibles predictores, donde $K \geq 1$. En algunos ejemplos, el codificador de vídeo puede generar una combinación completa de todos los predictores mediante el uso de vectores de movimiento vecinos disponibles. Como se muestra en la Figura 8, puede haber un total de $3 \times 2 \times 2 = 12$ combinaciones.

En la primera etapa, para cada combinación, el codificador de vídeo puede realizar una verificación de validación. Si MV0 es igual a MV1 y MV0 es igual a MV2, esta combinación no es válida; de cualquier otra manera, es válida. En la segunda etapa, el codificador de vídeo puede realizar una clasificación en base a la similitud de los parámetros. Por ejemplo, si el bloque actual usa el modo afín de seis parámetros como sigue, donde a, b, c, d, e y f son parámetros del modelo, el modelo de movimiento afín se puede representar de acuerdo con la Ecuación (3), que se reproduce más abajo.

$$\begin{cases} v_x = ax - by + c \\ v_y = dx + ey + f \end{cases} \quad (3)$$

Mediante el uso del modelo afín de seis parámetros, los tres vectores de movimiento de las esquinas se pueden representar como sigue:

$$\begin{cases} MV0_v_x = c \\ MV0_v_y = f \end{cases}$$

$$\begin{cases} MV1_v_x = a \times ancho + c \\ MV1_v_y = d \times ancho + f \end{cases} \quad (4)$$

$$\begin{cases} MV2_v_x = b \times ancho + c \\ MV2_v_y = e \times ancho + f \end{cases}$$

La Figura 9 es un diagrama conceptual que ilustra la evaluación del vector de movimiento afín, de acuerdo con una o más técnicas de esta divulgación. Con el fin de evaluar la exactitud del modelo, esta divulgación introduce un parámetro llamado diferencia estimada (ED). Al mismo tiempo, pueden usarse dos bloques vecinos de MV en el procedimiento de evaluación resaltados en los bloques vecinos 902H y 902I que se localizan la mitad en el ancho y la mitad en la altura como se muestra en la Figura 9. Por tanto hay:

$$\begin{cases} MVH_v_x = a \times ancho / 2 + c \\ MVH_v_y = d \times ancho / 2 + f \end{cases} \quad (5)$$

$$\begin{cases} MVI_v_x = b \times altura / 2 + c \\ MVI_v_y = e \times ancho / 2 + f \end{cases}$$

Entre todas las combinaciones, las primeras K combinaciones con menos ED pueden seleccionarse como el predictor final. El siguiente es un ejemplo de cálculo de ED:

$$\begin{aligned} \Delta a \times altura &= abs((MVI_v_x - MV0_v_x) - (MVH_v_x - MV0_v_x) \times 2) \times altura \\ \Delta b \times ancho &= abs((MVI_v_y - MV0_v_y) - (MVI_v_x - MV0_v_x) \times 2) \times ancho \quad ((\\ \Delta d \times altura &= abs((MV1_v_y - MV0_v_y) - (MVH_v_y - MV0_v_y) \times 2) \times altura \quad 6) \\ \Delta e \times ancho &= abs((MV2_v_x - MV0_v_x) - (MVI_v_y - MV0_v_y) \times 2) \times ancho \end{aligned}$$

El codificador de vídeo puede establecer la ED igual a la suma de los cuatro elementos anteriores.

$$ED = \Delta a + \Delta b + \Delta d + \Delta e \quad (7)$$

En algunos ejemplos, el codificador de vídeo puede realizar una clasificación en base a la similitud de vectores de movimiento afines. En un ejemplo, dados tres vectores de movimiento, el codificador de vídeo puede predecir el cuarto vector de movimiento mediante el uso de un modelo afín de seis parámetros. La diferencia de predicción se puede adicionar en la ED y las primeras combinaciones con la ED más pequeña se pueden elegir como candidatas de predicción de MV.

Los predictores de vector de movimiento se pueden generar a través de los otros predictores mediante el uso de un modelo afín de cuatro parámetros. Por ejemplo, dados los dos primeros MV reconstruidos, el codificador de vídeo puede generar un tercer predictor de MV mediante el uso del modelo afín de cuatro parámetros. Por ejemplo, el predictor de MV para MV2 se puede derivar en base a MV0 y MV1 del bloque actual mediante el uso de la Ecuación (2) anterior.

En algunos ejemplos, el predictor de vector de movimiento afín se puede generar a partir de los vectores de movimiento afín previamente codificados dentro del fotograma actual. En un ejemplo, se puede inicializar un conjunto de N (N>=0) vectores de movimiento afines al comienzo de cada fotograma, y después de codificar cada bloque afín, la lista se actualiza con los vectores de movimiento afines codificados recientemente y se señala un índice para indicar el predictor de movimiento afín elegido entre la lista. El codificador de vídeo puede usar código unario truncado o bandera más código unario truncado para codificar el índice.

En algunos ejemplos, un conjunto de K (K>= 0) parámetros de modelo afines se inicializa al comienzo de cada fotograma. Después de codificar cada bloque afín, el conjunto de parámetros se actualiza con los parámetros codificados del modelo afín. Por ejemplo, en el modelo de seis parámetros, el codificador de vídeo puede mantener una lista de N vectores, donde cada vector se representa por {a_i, b_i, c_i, d_i, e_i, f_i} con seis elementos. De manera similar, en el modo de cuatro parámetros, el codificador de vídeo puede mantener una lista de M vectores {a_j, b_j, c_j, d_j}. Tenga en cuenta que M y N pueden ser iguales o no.

En las técnicas mencionadas anteriormente, para el modo inter afín, el codificador de vídeo puede derivar el predictor del vector de movimiento de cada MV del modelo afín individualmente, mediante el uso de los MV de su posición vecina. De acuerdo con las realizaciones de la invención protegida, cuando el movimiento afín es usado por un bloque vecino, un codificador de vídeo puede usar el modelo de movimiento afín del bloque vecino para predecir todos los MV del modelo de movimiento afín del bloque actual, es decir, los predictores de MV0 y MV1 (y MV2 para modelos de seis parámetros) del modelo afín actual se extrapola del movimiento afín de un bloque vecino y luego codificar la MVD.

Los diferentes procedimientos de predicción mencionados anteriormente se pueden usar conjuntamente. Por ejemplo, se puede señalar una bandera o índice para indicar qué procedimiento de predicción de MV se usa. En algunos ejemplos, los predictores derivados mediante el uso de los diferentes métodos de predicción mencionados anteriormente se usan para generar una lista de candidatos a predictores de MV, y se usa una bandera o índice para indicar qué candidato se usa para predecir el modelo de movimiento afín actual.

Cuando se usa un modelo de movimiento afín de cuatro parámetros, pueden usarse "MV0 y MV1" o "MV0 y MV2" (v₀ y v₁ o v₀ y v₂ como se muestra en la Figura 8) para representar el movimiento afín de la CU/PU actual. Cuando el ancho y la altura de la CU/PU actual es diferente, se puede usar cierto tipo de regla para determinar qué par de

vectores de movimiento se usa.

En un ejemplo, cuando el ancho es mayor o igual que (o sólo mayor que) la altura o la relación de ancho y altura es mayor que un umbral, puede usarse el par de MV0 y MV1, de cualquier otra manera, puede usarse el par de MV0 y MV2. El umbral puede depender del tamaño del bloque o depender del ancho/altura.

Las técnicas pueden aplicarse tanto al modo de fusión afín como al modo inter afín, o aplicarse sólo en uno de ellos, por ejemplo, en el modo de fusión afín.

El codificador de vídeo puede usar una orden de verificación/evaluación particular para seleccionar un bloque vecino (por ejemplo, en modo de fusión). En algunos ejemplos, el codificador de vídeo puede usar el siguiente orden para verificar los bloques vecinos para el modo de fusión afín: Superior -> Izquierda -> Superior izquierda -> Superior derecha -> Inferior izquierda. Este orden corresponde a los bloques en la Figura 9 como D -> F -> A -> E -> G. Cuando los bloques vecinos no están disponibles o no son bloques codificados afines, el codificador de vídeo puede aplicar la verificación en el orden predefinido hasta que se verifican los cinco candidatos.

En algunos ejemplos, si no hay bloques de movimiento afín vecinos disponibles, el codificador de vídeo puede insertar ciertos modelos de movimiento afines predeterminados o predefinidos o precalculados como los candidatos para el modo de fusión. Los modelos que se insertan se pueden inicializar como el nivel de imagen y pueden actualizarse sobre la marcha.

En algunos ejemplos, si no hay modelos afines vecinos válidos, el codificador de vídeo puede realizar la inserción de modelos de movimiento afines predeterminados o predefinidos o precalculados después de verificar los bloques vecinos de acuerdo con el orden "Superior -> Izquierda -> Superior Izquierda -> Superior derecha -> Inferior izquierda".

En algunos ejemplos, el codificador de vídeo puede codificar un índice de fusión afín para indicar qué modelos afines vecinos se copian para el bloque actual y para codificar el índice puede usarse el unario truncado, o el unario, o el Golomb exponencial, o la palabra de código de la familia Golomb, o la concatenación de éstos.

Modelo afín conmutable de cuatro parámetros y seis parámetros derivado/inferido de otra información. En algunos ejemplos, el codificador de vídeo puede derivar el parámetro afín a partir de la información de dirección de interpredicción. Para cada bloque, si se codifica mediante el uso del intermodo, el índice del fotograma de referencia de predicción puede ser de refList0, o de refList1, o tanto de refList0 como de refList1. De acuerdo con una o más técnicas de esta divulgación, cuando se usa unipredicción (ya sea predicha a partir de refList0 o predicha a partir de refList1), un codificador de vídeo puede usar un modelo afín de seis parámetros en el que se codifican tres diferencias de vectores de movimiento en el flujo de bits. Cuando se usa la bipredicción (predicha a partir tanto de refList0 como de refList1), un codificador de vídeo puede usar un modelo afín de cuatro parámetros en el que se codifican dos diferencias de vectores de movimiento en el flujo de bits. En algunos de tales ejemplos, el codificador de vídeo puede omitir la codificación del elemento de sintaxis que indica explícitamente si se usa un modelo afín de cuatro parámetros o de seis parámetros para identificar uno o más bloques predictores de datos de vídeo para un bloque actual de datos de vídeo.

De acuerdo con una o más técnicas de esta divulgación, para el bloque de bipredicción, cuando L1ZeroMVDFlag está activado, un codificador de vídeo puede habilitar un modelo afín de seis parámetros para refList1 aunque no se transmita ninguna MVD. En este caso, el codificador de vídeo puede generar el predictor de movimiento compensado a través del modelo afín de seis parámetros establecido por los tres predictores de vector de movimiento.

En algunos ejemplos, el parámetro afín se puede derivar del bloque vecino. Si la mayoría de los bloques vecinos usan el modo afín de cuatro parámetros, el bloque actual también usa el modelo afín de cuatro parámetros. De manera similar, cuando la mayoría de los bloques vecinos usan un modelo afín de seis parámetros (el número de afines de seis parámetros es mayor que el de afines de cuatro parámetros), el bloque actual también usa un modelo afín de seis parámetros. Puede usarse un contador para calcular el número de bloques vecinos en cierto tamaño de unidad (para bloques de 4x4) para determinar el uso mayoritario afín vecino. Cuando no hay un modelo afín vecino, se usa el modelo afín de seis parámetros como modo por defecto (alternativamente, se usa el modelo afín de cuatro parámetros como modo por defecto). Cuando el número del modelo afín de cuatro parámetros es igual al del modelo de seis parámetros, se usa como por defecto el modelo afín de seis parámetros (alternativamente, se usa como por defecto el modelo afín de cuatro parámetros).

Determinación de fotogramas cruzados de banderas de modelos afines y vectores de movimiento. De acuerdo con una o más técnicas de esta divulgación, un codificador de vídeo puede usar los parámetros del modelo de movimiento afín de fotogramas cruzados en lugar de señalar explícitamente las bandera de parámetros afines (modo de cuatro o seis parámetros) o la información del vector de movimiento afín. En un ejemplo, el bloque actual hereda la bandera del modelo de parámetro afín del bloque colocado. El bloque colocado procede de la misma localización pero en la imagen previamente codificada en el mismo nivel temporal. El bloque colocado puede tener o no el mismo

tamaño de partición que el bloque actual. De acuerdo con una o más técnicas de esta divulgación, un codificador de vídeo puede verificar todos los subbloques (en la unidad de 4x4) en la región colocada, y la mayoría del modelo afín se usa para el bloque actual. Si no hay un modelo afín en la región colocada, el codificador de vídeo puede codificar explícitamente la bandera de conmutación de cuatro o seis parámetros. En algunos ejemplos, se usa como por defecto una afinidad de 6 (o 4) parámetros. En algunos ejemplos, para reducir la complejidad, el primer subbloque afín de la región colocada en el orden de escaneo de trama se verifica y hereda por el bloque actual.

En otro ejemplo, el bloque actual hereda los parámetros del modelo de movimiento afín {a, b, c, d, e, f} o {a, b, c, d} directamente del bloque colocado. El bloque colocado procede de la misma localización pero en la imagen previamente codificada con el mismo nivel temporal. El bloque colocado puede tener o no el mismo tamaño de partición que el bloque actual. De acuerdo con una o más técnicas de esta divulgación, un codificador de vídeo puede verificar todos los subbloques (en la unidad de 4x4) en la región colocada y el bloque actual hereda los parámetros del modelo de movimiento del área afín mayoritaria. Si no hay un modelo afín en la región colocada, el codificador de vídeo puede codificar explícitamente una bandera de conmutación de cuatro o seis parámetros. En algunos ejemplos, se usa como por defecto una afinidad de seis (o cuatro) parámetros. En algunos ejemplos, para reducir la complejidad, el primer subbloque afín de la región colocada en el orden de escaneo de trama se verifica y hereda por el bloque actual. En algunos ejemplos, puede usarse una combinación de los ejemplos anteriores juntos. Un codificador de vídeo puede codificar una bandera para indicar si tal herencia se usa o no en diferentes niveles, tal como la PU, el nivel de CU, el PPS o el SPS.

Compensación de movimiento afín dada la información de parámetro afín. En el procedimiento de reconstrucción, dados tres vectores de movimiento (por ejemplo, el vector de movimiento de la esquina en el bloque actual), se puede establecer un modelo afín de seis parámetros al resolver la Ecuación (4). Dado el modelo de seis parámetros, el vector de movimiento por píxel se puede calcular al sustituir la posición del píxel (x,y) en la Ecuación (3). Para reducir la complejidad de la compensación de movimiento, puede usarse un vector de movimiento para cada subbloque KxK, donde K es un entero igual o mayor que 1. El vector de movimiento representativo se puede calcular mediante el uso de la posición del píxel superior izquierdo dentro del subbloque KxK, o mediante el uso de la posición central del subbloque KxK. El tamaño K puede señalizarse explícitamente o establecerse como un valor por defecto o calcularse sobre la marcha en base a si el grupo de píxeles comparte el mismo vector de movimiento.

Codificación de vector de movimiento afín. Los predictores de los vectores de movimiento vecinos válidos (en términos de validación del modelo afín) y desduplicados pueden usarse para identificar/predecir el vector de movimiento afín actual. Los predictores de los últimos vectores de movimiento afines codificados previamente desduplicados pueden mantenerse para identificar/predecir el vector de movimiento afín actual. El número de predictores puede ser K, donde K es un entero igual o mayor que 1. Estos predictores forman una lista de predictores afines. K puede predefinirse o señalizarse en el flujo de bits.

En algunos ejemplos, puede usarse una combinación de ambas técnicas anteriores para mantener la lista de predictores. Por ejemplo, un codificador de vídeo puede usar predictores de los vectores de movimiento vecinos válidos (en términos de validación del modelo afín) y desduplicados junto con predictores de los últimos vectores de movimiento afines codificados previamente desduplicados para identificar/predecir el vector de movimiento afín actual.

El codificador de vídeo puede señalar explícitamente un índice de predictor en el flujo de bits para indicar el uso del predictor. Tres MVD pueden decodificarse en el caso del modelo de seis parámetros, mientras que se pueden codificar dos MVD en el caso del modelo de cuatro parámetros.

La MVD puede usar un procedimiento de binarización diferente al de la codificación de MVD tradicional. En un ejemplo, la MVD afín se codifica mediante el uso de un modelado de contexto separado. En otro ejemplo, la codificación de MVD afín comparte el mismo modelado de contexto de codificación de MVD con la codificación inter-MVD tradicional (es decir, como en HEVC).

La MVD puede usar un procedimiento de binarización diferente para cada MVD en base a la localización relativa en el bloque con un modelo afín de cuatro parámetros o seis parámetros. En un ejemplo, la MVD afín se puede codificar mediante el uso de un modelado de contexto diferente en base a la localización relativa en el bloque con un modelo afín o bien de cuatro parámetros o de seis parámetros.

Se puede señalar una bandera para indicar si las MVD en ambas direcciones (direcciones X e Y) son cero para uno o todos los vectores de movimiento afines para mejorar aún más la codificación del vector de movimiento. Si tal una bandera (AllZeroFlag) es 1, se introduce una nueva codificación de MVD para codificar conjuntamente MVD_x y MVD_y. Específicamente, si AllZeroFlag es 1, se infiere que tanto MVD_x como MVD_y son cero; de cualquier otra manera, si MVD_x es cero, MVD_y debe ser distinto de cero. En este caso, se codifica $\text{abs}(\text{MVD}_y) - 1$. En otras palabras, para cada vector de movimiento, se señala una bandera AllZeroFlag seguida de dos codificaciones de MVD si AllZeroFlag es cero. Para afines de cuatro parámetros, para cada lista, se codifican dos AllZeroFlags; mientras que para afines de seis parámetros, para cada lista, se codifican tres AllZeroFlags.

En algunos ejemplos, AllZeroFlag se puede ampliar y representar todas las MVD cero en ambas listas de referencia en la bipredicción. Por ejemplo, en el afín de cuatro parámetros, se codifican dos AllZeroFlags para dos listas de referencia; en el afín de seis parámetros, se codifican en total tres AllZeroFlags para dos listas de referencia.

- 5 La Figura 10 ilustra un ejemplo de compensación de movimiento de bloques solapados (OBMC). Propuesto en el desarrollo de H.263, la OBMC se realiza en un bloque de 8x8 y los vectores de movimiento de dos bloques de 8x8 vecinos conectados se usan para un bloque actual. Por ejemplo, para un primer bloque de 8x8 en el macrobloque actual, además del vector de movimiento del primer bloque de 8x8, los vectores de movimiento vecinos superior e izquierdo del primer bloque de 8x8 también se aplican para generar dos bloques de predicción adicionales. De manera similar, para un segundo bloque de 8x8 en el macrobloque actual, además del vector de movimiento del segundo bloque de 8x8, los vectores de movimiento vecino superior y derecho del segundo bloque de 8x8 también se aplican para generar dos bloques de predicción adicionales. Por ejemplo, en el ejemplo de la Figura 10, los vectores de movimiento del bloque 1004A y el bloque 1004B pueden usarse para generar bloques de predicción adicionales para el bloque 1002A de 8x8 del macrobloque 1000 de 16x16, y los vectores de movimiento del bloque 1006A y el bloque 1006B pueden usarse para generar bloques de predicción adicionales para el bloque 1002B de 8x8 del macrobloque 1000. De esta manera, cada píxel en el bloque de 8x8 actual puede tener tres bloques de predicción y el promedio ponderado de estos tres valores de predicción puede usarse como el bloque de predicción final.
- 10
- 15
- 20 Cuando un bloque vecino no está codificado o codificado como intra (es decir, el bloque vecino no tiene un vector de movimiento disponible), el vector de movimiento del bloque de 8x8 actual se usa como vector de movimiento vecino. Mientras tanto, para el tercer y cuarto bloque de 8x8 del macrobloque actual (como se muestra en la Figura 10), el bloque vecino inferior no siempre se usa. Por ejemplo, como se muestra en el ejemplo de la Figura 10, el vector de movimiento del bloque 1008B no se usa para generar un bloque de predicción adicional para el bloque 1002C de 8x8 porque se considera que el bloque 1008B no está codificado y el vector de movimiento del bloque 1010B no se usa para generar un bloque de predicción adicional para el bloque 1002D de 8x8 porque se considera que el bloque 1010B no está codificado. En otras palabras, para cada macrobloque, no se usará ninguna información de movimiento de los macrobloques inferiores para reconstruir los píxeles del macrobloque actual durante la OBMC.
- 25
- 30 Las Figuras 11A y 11B son diagramas conceptuales que ilustran la OBMC en HEVC. En HEVC, también se propuso la OBMC para suavizar el límite de PU en la publicación de solicitud de patente de Estados Unidos Núm. 2013/0128974A1 y publicación de solicitud de patente de Estados Unidos Núm. 2012/0177120A1. Un ejemplo del procedimiento propuesto se ilustra en las Figuras 11A y 11B. En las Figuras 11A y 11B, cada una de las regiones blancas es una primera PU 1102 (PU0) y cada una de las regiones sombreadas es una segunda PU 1104 (PU1).
- 35 Cuando una CU contiene dos (o más) PU, las líneas/columnas cercanas al límite de la PU se suavizan mediante la OBMC. Para los píxeles marcados con "A" o "B" en PU0 1102 o PU1 1104, se generan dos valores de predicción, por ejemplo, al aplicar los vectores de movimiento de PU0 y PU1 respectivamente, y el promedio ponderado de ellos se usa como la predicción final.
- 40 Las Figuras 12A y 12B son diagramas conceptuales que ilustran los subbloques donde se puede aplicar la OBMC. En el software de referencia del modelo de exploración conjunta (JEM) (disponible en <https://jvet.hhi.fraunhofer.de/>), se aplica la OBMC a nivel de sub-PU. La OBMC se realiza para todos los límites de bloque con compensación de movimiento (MC), excepto los límites derecho e inferior de una CU. Además, se aplica tanto para componentes de luma como de croma. En HEVC, un bloque de MC se corresponde a una PU. En el JEM, cuando una PU se codifica con el modo sub-PU, cada subbloque de la PU es un bloque de MC. Para procesar los límites de la CU/PU de manera uniforme, la OBMC se realiza a nivel de subbloque para todos los límites de bloque de MC, donde el tamaño del subbloque se establece igual a 4x4, como se ilustra en las Figuras 12A y 12B.
- 45
- 50 Cuando la OBMC se aplica al subbloque actual, además de los vectores de movimiento actuales, los vectores de movimiento de cuatro subbloques vecinos conectados también se usan para derivar el bloque de predicción para el subbloque actual si están disponibles y no son idénticos al vector de movimiento actual. Estos múltiples bloques de predicción en base a los múltiples vectores de movimiento se ponderan para generar la señal de predicción final del subbloque actual.
- 55 Los bloques de predicción en base a los vectores de movimiento de un subbloque vecino pueden denotarse como P_N , con n que indica un índice para los subbloques vecinos superior, inferior, izquierda y derecha. El bloque de predicción en base a los vectores de movimiento de un bloque actual se puede denotar como P_C . Cuando P_N pertenece a la misma PU que P_C (por tanto contiene la misma información de movimiento), la OBMC no se realiza desde P_N . De cualquier otra manera, cada píxel de P_N se adiciona al mismo píxel en P_C , es decir, cuatro filas/columnas de P_N se agregan a P_C . Los factores de ponderación $\{1/4, 1/8, 1/16, 1/32\}$ se usan para P_N y los factores de ponderación $\{3/4, 7/8, 15/16, 31/32\}$ se usan para P_C . La excepción son los bloques de MC pequeños (es decir, cuando el tamaño de la PU es igual a 8x4, 4x8 o una PU está codificada con el modo ATMVP), para los cuales solo dos filas/columnas de P_N se adicionan a P_C . En este caso, pueden usarse los factores de ponderación $\{1/4, 1/8\}$ para P_N y los factores de ponderación $\{3/4, 7/8\}$ se usan para P_C . Para P_N generados en base a los vectores de movimiento de subbloques vecinos verticalmente (horizontalmente), los píxeles en la misma fila (columna) de P_N se adicionan a P_C con el mismo factor de ponderación. Tenga en cuenta que para los límites de PU, se puede aplicar la
- 60
- 65

OBMC en cada lado del límite. Tal como en las Figuras 12A y 12B, la OBMC se puede aplicar a lo largo del límite entre PU1 y PU2 dos veces. Primero, se aplica la OBMC con el MV de PU2 a los bloques sombreados a lo largo del límite dentro de PU1. Segundo, se aplica la OBMC con el MV de PU1 a los bloques sombreados a lo largo del límite dentro de PU2. Por el contrario, la OBMC solo se puede aplicar a un lado de los límites de la CU porque al codificar la CU actual, no se puede cambiar las CU que ya se han codificado.

La Figura 13 es un diagrama de flujo que ilustra un procedimiento de ejemplo para realizar una compensación de movimiento afín mediante un codificador de vídeo (por ejemplo, durante un procedimiento de codificación de vídeo), de acuerdo con una o más técnicas de esta divulgación. Para fines de ejemplo y explicación, el procedimiento de la Figura 13 se describe con respecto al codificador de vídeo 20 de las Figuras 1 y 2.

El codificador de vídeo 20 puede recibir un bloque actual de datos de vídeo a codificar (1302). Por ejemplo, el codificador de vídeo 20 puede recibir, desde la fuente de vídeo 18, valores de píxeles en bruto (por ejemplo, RGB, CMYK, YUV, etc.) para una imagen actual de datos de vídeo que incluye el bloque actual de datos de vídeo. La unidad de partición 48 de la unidad de selección de modo 40 del codificador de vídeo 20 puede dividir la imagen actual en una pluralidad de bloques, uno de los cuales puede ser el bloque actual.

El codificador de vídeo 20 puede determinar codificar el bloque actual de datos de vídeo mediante el uso de predicción de movimiento afín (1304). Por ejemplo, la unidad de selección de modo 40 puede determinar codificar el bloque actual de datos de vídeo mediante el uso del modo de interpredicción y seleccionar el modelo de movimiento afín como modo de predicción de la información de movimiento. La unidad de selección de modo 40 puede determinar el uso del modo de interpredicción en base a una amplia variedad de factores, tales como un tipo de fotograma de la imagen actual (por ejemplo, un fotograma P, un fotograma I, un fotograma B, etc.), y qué modo de predicción da como resultado el menor costo de optimización de la tasa de distorsión (RDO).

El codificador de vídeo 20 puede codificar una indicación de que el bloque actual se codifica mediante el uso de predicción de movimiento afín (1306). Por ejemplo, la unidad de selección de modo 40 puede hacer que la unidad de codificación de entropía 56 del codificador de vídeo 20 codifique, en un flujo de bits de vídeo, uno o más elementos de sintaxis que indican que el bloque actual se codifica mediante el uso del modo de interpredicción, uno o más elementos de sintaxis que indican que el modelo de movimiento afín es el modo de predicción de información de movimiento para el bloque actual, y/o uno o más elementos de sintaxis que indican que el bloque actual se codifica mediante el uso del modo de interpredicción y el modelo de movimiento afín es el modo de predicción de información de movimiento para el bloque actual.

El codificador de vídeo 20 puede determinar los valores de los vectores de movimiento de un modelo de movimiento afín del bloque actual de datos de vídeo (1308). Por ejemplo, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 del codificador de vídeo 20 pueden identificar un bloque predictor de datos de vídeo que tiene valores de píxeles que coinciden estrechamente con los valores de píxeles del bloque actual de datos de vídeo. La unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 pueden determinar dos o más vectores de movimiento que representan una transformación afín entre el bloque actual de datos de vídeo y el bloque predictor de datos de vídeo.

Como se divulgó anteriormente, en algunos ejemplos, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 siempre pueden usar un modelo de movimiento afín de cuatro parámetros que incluye dos vectores de movimiento para identificar el bloque predictor. De manera similar, en algunos ejemplos, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 siempre pueden usar un modelo de movimiento afín de seis parámetros que incluye tres vectores de movimiento para identificar el bloque predictor. En aún otros ejemplos, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 pueden usar selectivamente o bien un modelo de movimiento afín de cuatro parámetros que incluye dos vectores de movimiento (por ejemplo, v_0 y v_1 de la Figura 8, también denominados como MV0 y MV1) o un modelo de movimiento afín de seis parámetros que incluye tres vectores de movimiento (por ejemplo, v_0 , v_1 , y v_2 de la Figura 8, también denominados como MV0, MV1 y MV2) para identificar el bloque predictor.

En algunos ejemplos, el codificador de vídeo 20 puede codificar una indicación de si el bloque actual se codifica mediante el uso de un modelo de cuatro parámetros o un modelo de seis parámetros. Por ejemplo, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 pueden hacer que la unidad de codificación de entropía 56 codifique, en un flujo de bits de vídeo codificado, un elemento de sintaxis que indica si el modelo de movimiento afín para el bloque actual de datos de vídeo comprende un modelo de cuatro parámetros o un modelo de seis parámetros. En algunos ejemplos, la unidad de codificación de entropía 56 puede codificar el elemento de sintaxis en uno o más de un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un encabezado del segmento al que hace referencia el bloque actual de datos de vídeo. En algunos ejemplos, la unidad de codificación de entropía 56 puede codificar el elemento de sintaxis en el nivel de unidad de codificación (CU) de una CU que incluye el bloque actual de datos de vídeo.

El codificador de vídeo 20 puede seleccionar, para el bloque actual de datos de vídeo, un bloque vecino de datos de

vídeo que tiene un modelo de movimiento afín (1310). Por ejemplo, cuando se codifica el bloque actual 800 de la Figura 8, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 pueden evaluar los bloques 802A-802G de la Figura 8 en un orden particular y seleccionar el primer bloque, en el orden particular, que se codifica mediante el uso de compensación de movimiento afín (por ejemplo, el primer bloque que tiene un modelo de movimiento afín disponible) como el bloque vecino de datos de vídeo seleccionado. En algunos ejemplos, el bloque actual de datos de vídeo puede codificarse mediante el uso del modo inter afín. En algunos ejemplos, el bloque vecino de datos de vídeo seleccionado puede codificarse mediante el uso del modo inter afín (por ejemplo, AF_INTER) o el modo de fusión afín (por ejemplo, AF_MERGE).

El codificador de vídeo 20 puede obtener los valores de predictores de vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo seleccionado (1312). Por ejemplo, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 pueden obtener los valores del modelo de movimiento afín del bloque vecino de datos de vídeo seleccionado desde una memoria o dispositivo de almacenamiento del codificador de vídeo 20, tal como la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 pueden deformar los valores del modelo de movimiento afín del bloque vecino de datos de vídeo seleccionado a la posición del bloque actual para derivar los valores de los predictores. En otras palabras, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 pueden extrapolar los valores de los predictores a partir de los valores del modelo de movimiento afín del bloque vecino de datos de vídeo seleccionado. Como ejemplo, donde el bloque vecino seleccionado es el bloque 802F de la Figura 8, el codificador de vídeo 20 puede obtener valores de una pluralidad de vectores de movimiento del bloque 802F (por ejemplo, valores de los CPMV del bloque 802F) y deformar los valores de la pluralidad de vectores de movimiento del bloque 802F a la posición del bloque actual 800. Como otro ejemplo, donde el bloque vecino seleccionado es el bloque 802F de la Figura 8, el codificador de vídeo 20 puede usar los valores de la pluralidad de vectores de movimiento del bloque 802F (por ejemplo, valores de los CPMV del bloque 802F) como los predictores.

El codificador de vídeo 20 puede codificar, en un flujo de bits de vídeo codificado, una representación de las diferencias entre los valores de los vectores de movimiento de un modelo de movimiento afín para el bloque actual de datos de vídeo y los valores de los predictores (1314). Por ejemplo, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 pueden determinar, para cada vector de movimiento respectivo del modelo de movimiento afín del bloque actual, un valor de diferencia de vector de movimiento (MVD) respectivo que representa la diferencia entre el valor del vector de movimiento respectivo del modelo de movimiento afín del bloque actual y el valor de un predictor correspondiente derivado de los vectores de movimiento del modelo de movimiento afín del bloque vecino seleccionado. Como ejemplo, donde los valores de los vectores de movimiento del modelo de movimiento afín del bloque actual son MV0 y MV1 y los valores de los predictores derivados de los vectores de movimiento del modelo de movimiento afín del bloque vecino seleccionado son MVP0 y MVP1, la unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 pueden determinar un primer valor de MVD como una diferencia entre MV0 y MVP0, y determinar un segundo valor de MVD como una diferencia entre MV1 y MVP1. La unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 pueden hacer que la unidad de codificación de entropía 56 codifique, en el flujo de bits de vídeo codificado, uno o más elementos de sintaxis que representan los valores de las MVD determinadas.

En algunos ejemplos, el codificador de vídeo 20 puede codificar además, en el flujo de bits de vídeo codificado, datos residuales que representan diferencias de píxeles entre el bloque actual y un bloque predictor identificado por el modelo de movimiento afín del bloque actual. El codificador de vídeo 20 puede implementar un bucle decodificador para reconstruir los valores de píxeles del bloque actual (por ejemplo, para su uso al predecir bloques futuros). Por ejemplo, el codificador de vídeo 20 puede identificar el bloque predictor en base al modelo de movimiento afín para el bloque actual, obtener los valores de píxeles del bloque predictor de la memoria de imágenes de referencia 64 y adicionar los valores residuales a los valores de píxeles del bloque predictor para reconstruir los valores de píxeles del bloque actual.

La Figura 14 es un diagrama de flujo que ilustra un procedimiento de ejemplo para realizar una compensación de movimiento afín mediante un decodificador de vídeo (por ejemplo, durante un procedimiento de codificación de vídeo), de acuerdo con una o más técnicas de esta divulgación. Para fines de ejemplo y explicación, el procedimiento de la Figura 14 se describe con respecto al decodificador de vídeo 30 de las Figuras 1 y 3.

El decodificador de vídeo 30 puede decodificar una indicación de que el bloque actual se codifica mediante el uso de predicción de movimiento afín (1402). Por ejemplo, la unidad de codificación de entropía 70 puede decodificar a partir de un flujo de bits de vídeo, uno o más elementos de sintaxis que indican que el bloque actual se codifica mediante el uso del modo de interpredicción, uno o más elementos de sintaxis que indican que el modelo de movimiento afín es el modo de predicción de información de movimiento para el bloque actual, y/o uno o más elementos de sintaxis que indican que el bloque actual se codifica mediante el uso del modo de interpredicción y el modelo de movimiento afín es el modo de predicción de información de movimiento para el bloque actual. La unidad de decodificación de entropía 70 puede proporcionar los valores de los elementos de sintaxis decodificados a la unidad de compensación de movimiento 72.

El decodificador de vídeo 30 puede seleccionar, para el bloque actual de datos de vídeo, un bloque vecino de datos de vídeo que tiene un modelo de movimiento afín (1404). Por ejemplo, cuando se decodifica el bloque actual 800 de la Figura 8, la unidad de compensación de movimiento 72 puede evaluar los bloques 802A-802G de la Figura 8 en un orden particular y seleccionar el primer bloque, en el orden particular, que se codifica mediante el uso de compensación de movimiento afín (por ejemplo, el primer bloque que tiene un modelo de movimiento afín disponible) como el bloque vecino de datos de vídeo seleccionado. En algunos ejemplos, el bloque actual de datos de vídeo puede codificarse mediante el uso del modo inter afín. En algunos ejemplos, el bloque vecino de datos de vídeo seleccionado puede codificarse mediante el uso del modo inter afín (por ejemplo, AF_INTER) o el modo de fusión afín (por ejemplo, AF_MERGE).

El decodificador de vídeo 30 puede obtener los valores de predictores derivados a partir de vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo seleccionado (1406). Por ejemplo, la unidad de compensación de movimiento 72 puede obtener los valores del modelo de movimiento afín del bloque vecino de datos de vídeo seleccionado desde una memoria o dispositivo de almacenamiento del decodificador de vídeo 30, tal como la memoria de imágenes de referencia 82. La unidad de compensación de movimiento 72 puede deformar los valores del modelo de movimiento afín del bloque vecino de datos de vídeo seleccionado a la posición del bloque actual para derivar los valores de los predictores. En otras palabras, la unidad de compensación de movimiento 72 puede extrapolar los valores de los predictores a partir de los valores del modelo de movimiento afín del bloque vecino de datos de vídeo seleccionado. Como ejemplo, donde el bloque vecino seleccionado es el bloque 802F de la Figura 8, el decodificador de vídeo 30 puede obtener valores de una pluralidad de vectores de movimiento del bloque 802F (por ejemplo, valores de los CPMV del bloque 802F) y deformar los valores de la pluralidad de vectores de movimiento del bloque 802F a la posición del bloque actual 800. Como otro ejemplo, donde el bloque vecino seleccionado es el bloque 802F de la Figura 8, el decodificador de vídeo 30 puede usar los valores de la pluralidad de vectores de movimiento del bloque 802F (por ejemplo, valores de los CPMV del bloque 802F) como los predictores.

El decodificador de vídeo 30 puede codificar, a partir de un flujo de bits de vídeo codificado, una representación de las diferencias entre los valores de los vectores de movimiento de un modelo de movimiento afín para el bloque actual de datos de vídeo y los valores de los predictores (1408). Por ejemplo, la unidad de decodificación de entropía 70 puede decodificar, a partir del flujo de bits de vídeo codificado, elementos de sintaxis que representan los valores de diferencias entre el valor del vector de movimiento respectivo del modelo de movimiento afín del bloque actual y el valor de un predictor correspondiente derivado a partir de los vectores de movimiento del modelo de movimiento afín del bloque vecino seleccionado. Como ejemplo, donde los valores de los vectores de movimiento del modelo de movimiento afín del bloque actual son MV0 y MV1 y los valores de los predictores derivados de los vectores de movimiento del modelo de movimiento afín del bloque vecino seleccionado son MVP0 y MVP1, la unidad de decodificación de entropía 70 puede decodificar elementos de sintaxis que representan el valor de un primer valor de la MVD y un segundo valor de la MVD, el primer valor de la MVD es una diferencia entre MV0 y MVP0 y el segundo valor de MVD es una diferencia entre MV1 y MVP1. La unidad de decodificación de entropía 70 puede proporcionar los valores de los elementos de sintaxis decodificados a la unidad de compensación de movimiento 72.

El decodificador de vídeo 30 puede determinar los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo en base a los valores de los predictores y las diferencias decodificadas (1410). Por ejemplo, la unidad de compensación de movimiento 72 puede agregar el valor de MVP0 al valor del primer valor de MVD para determinar el valor de MV0 y adicionar el valor de MVP1 al valor del segundo valor de MVD para determinar el valor de MV1.

El decodificador de vídeo 30 puede determinar, en base a los valores determinados de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo, un bloque predictor de datos de vídeo (1412). Por ejemplo, la unidad de compensación de movimiento 72 puede obtener, a partir de la memoria de imágenes de referencia 82, los valores de píxeles del bloque predictor identificado por el modelo de movimiento afín para el bloque actual de datos de vídeo.

El decodificador de vídeo 30 puede reconstruir el bloque actual de datos de vídeo en base al bloque predictor de datos de vídeo (1414). Por ejemplo, la unidad de decodificación de entropía 70 puede decodificar, a partir del flujo de bits de vídeo codificado, datos residuales que representan diferencias de píxeles entre el bloque actual y un bloque predictor identificado por el modelo de movimiento afín del bloque actual. La unidad de compensación de movimiento 72 puede adicionar los valores residuales a los valores de píxeles del bloque predictor para reconstruir los valores de píxeles del bloque actual.

Debe reconocerse que, en función del ejemplo, ciertos actos o eventos de cualquiera de las técnicas que se describen en la presente memoria pueden realizarse en una secuencia diferente, pueden adicionarse, fusionarse u omitirse por completo (por ejemplo, no todos los actos o eventos que se describen son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse concurrentemente, por ejemplo, a través de procesamiento de multihilos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, microprograma o cualquiera de sus combinaciones. Si se implementan en software, las funciones pueden almacenarse en o transmitirse como una o más instrucciones o código en un medio legible por ordenador y que se ejecutan mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible, tal como los medios de almacenamiento de datos o los medios de comunicación, que incluyen cualquier medio que facilite transferir un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medios de almacenamiento legibles por ordenador tangibles que no son transitorios o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder por uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A manera de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EPROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnéticos, memoria flash o cualquier otro medio que pueda utilizarse para almacenar el código del programa deseado en forma de instrucciones o estructuras de datos y al que puede accederse por un ordenador. Además, cualquier conexión se denomina apropiadamente como un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota mediante el uso de un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de suscriptor digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, pero en cambio se dirigen a medios de almacenamiento tangibles y no transitorios. Disco magnético y disco óptico, como se usa en la presente memoria, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco Blu-ray, donde los discos suelen reproducir datos magnéticamente, mientras que los discos reproducen datos ópticamente con láser. Las combinaciones de lo anterior también deben incluirse dentro del ámbito de los medios legibles por ordenador.

Las instrucciones pueden ejecutarse por uno o más procesadores, tal como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA) u otro circuito integrado lógico o discreto equivalente. En consecuencia, el término "procesador", como se usa en la presente memoria puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en la presente memoria. Además, en algunos aspectos, la funcionalidad descrita en la presente memoria puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y decodificación, o incorporarse en un códec combinado. Asimismo, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen varios componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización por diferentes unidades de hardware. Más bien, como se describió anteriormente, varias unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se describió anteriormente, junto con software y/o microprograma adecuados.

Se han descrito varios ejemplos.

REIVINDICACIONES

1. Un procedimiento para decodificar datos de vídeo, el procedimiento que comprende:

5 obtener, mediante uno o más procesadores de un decodificador de vídeo y para un bloque actual de una imagen actual de datos de vídeo, valores de los vectores de movimiento de un modelo de movimiento afín de un bloque vecino de datos de vídeo;
 decodificar, mediante el uno o más procesadores y a partir de un flujo de bits de vídeo codificado, un primer elemento de sintaxis que se incluye en un conjunto de parámetros de secuencia, SPS, al que hace referencia
 10 la imagen actual, en el que un primer valor del elemento de sintaxis indica que la compensación de movimiento para las imágenes que hacen referencia al SPS se puede realizar mediante el uso de un modelo de movimiento afín de cuatro parámetros definido por dos vectores de movimiento, y en el que un segundo valor del elemento de sintaxis indica que la compensación de movimiento para imágenes que hacen referencia al SPS se puede realizar mediante el uso o bien de un modelo de movimiento afín de cuatro parámetros definido por dos vectores de movimiento o un modelo de movimiento afín de seis parámetros definido por tres vectores de movimiento;
 15 en respuesta al primer elemento de sintaxis que tiene el segundo valor:
 decodificar, mediante el uno o más procesadores y a partir del flujo de bits de vídeo codificado, un segundo elemento de sintaxis en el que un primer valor del segundo elemento de sintaxis indica que la compensación de movimiento para el bloque actual se va a realizar mediante el uso de un modelo de movimiento afín de cuatro parámetros definido por dos vectores de movimiento, y en el que un segundo valor del segundo elemento de sintaxis indica que la compensación de movimiento para el bloque actual se va a realizar mediante el uso de un modelo de movimiento afín de seis parámetros definido por tres vectores de movimiento; y
 20 derivar, mediante el uno o más procesadores y a partir de los valores de los vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo, valores de predictores para los vectores de movimiento de un modelo de movimiento afín del bloque actual de datos de vídeo indicado por el segundo elemento de sintaxis;
 decodificar, mediante el uno o más procesadores y a partir del flujo de bits de vídeo codificado, una representación de las diferencias entre los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo y los valores de los predictores;
 30 determinar, mediante el uno o más procesadores, cada uno de los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo a partir de los valores de los predictores y las diferencias decodificadas;
 35 determinar, en base a los valores determinados de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo, un bloque predictor de datos de vídeo; y
 reconstruir el bloque actual de datos de vídeo en base al bloque predictor de datos de vídeo.

2. El procedimiento de la reivindicación 1, en el que el bloque actual de datos de vídeo se decodifica mediante el uso del modo inter afín y preferentemente en el que el bloque vecino de datos de vídeo se decodifica mediante el uso del modo inter afín o el modo de fusión afín.

3. El procedimiento de la reivindicación 1, en el que el bloque vecino de vídeo comprende un bloque vecino de datos de vídeo seleccionado, y en el que obtener los valores de los vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo seleccionado comprende:

45 evaluar, en un orden predefinido, bloques vecinos de datos de vídeo del bloque actual de datos de vídeo; y seleccionar un primer bloque vecino de datos de vídeo de la pluralidad de bloques vecinos de datos de vídeo decodificados mediante el uso de compensación de movimiento afín como el bloque vecino de datos de vídeo seleccionado y preferentemente en el que decodificar el segundo elemento de sintaxis comprende decodificar el elemento de sintaxis de una unidad de codificación, CU, que incluye el bloque actual de datos de vídeo.

4. Un procedimiento para codificar datos de vídeo, el procedimiento que comprende:

55 determinar, mediante uno o más procesadores de un codificador de vídeo, si la compensación de movimiento para un bloque actual de datos de vídeo se va a realizar mediante el uso de un modelo afín de cuatro parámetros definido por dos vectores de movimiento o un modelo afín de seis parámetros definido por tres vectores de movimiento;
 determinar, mediante el uno o más procesadores de un codificador de vídeo, los valores de los vectores de movimiento de un modelo de movimiento afín de un bloque actual de datos de vídeo, los vectores de movimiento del modelo de movimiento afín que identifican un bloque predictor de datos de vídeo para el bloque actual de datos de vídeo;
 60 obtener, mediante el uno o más procesadores, valores de los vectores de movimiento de un modelo de movimiento afín de un bloque vecino de datos de vídeo;
 derivar, mediante el uno o más procesadores y a partir de los valores de los vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo, valores de predictores para cada uno de los

vectores de movimiento del modelo de movimiento afin del bloque actual de datos de vídeo; y
codificar, mediante el uno o más procesadores y en un flujo de bits de vídeo codificado, una representación
de las diferencias entre los valores de los vectores de movimiento del modelo de movimiento afin para el
bloque actual de datos de vídeo y los valores de los predictores;

codificar, mediante el uno o más procesadores y en un flujo de bits de vídeo codificado, un primer elemento
de sintaxis que se incluye en un conjunto de parámetros de secuencia, SPS, al que hace referencia la imagen
actual, en el que un primer valor del elemento de sintaxis indica que la compensación de movimiento para las
imágenes que hacen referencia al SPS se puede realizar mediante el uso de un modelo de movimiento afin
de cuatro parámetros definido por dos vectores de movimiento, y en el que un segundo valor del primer
elemento de sintaxis indica que la compensación de movimiento para imágenes que hacen referencia al SPS
se puede realizar mediante el uso o bien de un modelo de movimiento afin de cuatro parámetros definido por
dos vectores de movimiento o un modelo de movimiento afin de seis parámetros definido por tres vectores de
movimiento; y

cuando el primer elemento de sintaxis tiene el segundo valor, codificar mediante el uno o más procesadores y
en el flujo de bits de vídeo codificado, un segundo elemento de sintaxis en el que un primer valor del segundo
elemento de sintaxis indica que la compensación de movimiento para el bloque actual se va a realizar
mediante el uso de un modelo de movimiento afin de cuatro parámetros definido por dos vectores de
movimiento, y en el que un segundo valor del segundo elemento de sintaxis indica que la compensación de
movimiento para el bloque actual se va a realizar mediante el uso de un modelo de movimiento afin de seis
parámetros definido por tres vectores de movimiento.

5. El procedimiento de la reivindicación 4, en el que el bloque actual de datos de vídeo se codifica mediante el uso
del modo inter afin y preferentemente en el que el bloque vecino de datos de vídeo se codifica mediante el uso
del modo inter afin o el modo de fusión afin.

6. El procedimiento de la reivindicación 4, en el que el bloque vecino de vídeo comprende un bloque vecino de
datos de vídeo seleccionado, y en el que obtener los valores de los vectores de movimiento del modelo de
movimiento afin del bloque vecino de datos de vídeo seleccionado comprende:

evaluar, en un orden predefinido, bloques vecinos de datos de vídeo del bloque actual de datos de vídeo; y
seleccionar un primer bloque vecino de datos de vídeo de la pluralidad de bloques vecinos de datos de vídeo
decodificados mediante el uso de compensación de movimiento afin como el bloque vecino de datos de vídeo
seleccionado y preferentemente en el que codificar el segundo elemento de sintaxis comprende codificar el
elemento de sintaxis en una unidad de codificación, CU, que incluye el bloque actual de datos de vídeo.

7. Un dispositivo (30) para decodificar un bloque de datos de vídeo, el dispositivo que comprende:

una memoria (82) que se configura para almacenar los datos de vídeo; y
una o más unidades de procedimiento que se implementan en circuitos y que se configuran para:

obtener, para un bloque actual de una imagen actual de datos de vídeo, valores de los vectores de
movimiento de un modelo de movimiento afin de un bloque vecino de datos de vídeo;
decodificar, a partir de un flujo de bits de vídeo codificado, un primer elemento de sintaxis que se incluye
en un conjunto de parámetros de secuencia, SPS, al que hace referencia la imagen actual, en el que un
primer valor del primer elemento de sintaxis indica que la compensación de movimiento para las
imágenes que hacen referencia al SPS se puede realizar mediante el uso de un modelo de movimiento
afin de cuatro parámetros definido por dos vectores de movimiento, y en el que un segundo valor del
primer elemento de sintaxis indica que la compensación de movimiento para imágenes que hacen
referencia al SPS se puede realizar mediante el uso o bien de un modelo de movimiento afin de cuatro
parámetros definido por dos vectores de movimiento o un modelo de movimiento afin de seis parámetros
definido por tres vectores de movimiento;
en respuesta al primer elemento de sintaxis que tiene el segundo valor:

decodificar, mediante el uno o más procesadores y a partir del flujo de bits de vídeo codificado, un
segundo elemento de sintaxis, en el que un primer valor del segundo elemento de sintaxis indica que
la compensación de movimiento para el bloque actual se va a realizar mediante el uso de un modelo
de movimiento afin de cuatro parámetros definido por dos vectores de movimiento, y en el que un
segundo valor del segundo elemento de sintaxis indica que la compensación de movimiento para el
bloque actual se va a realizar mediante el uso de un modelo de movimiento afin de seis parámetros
definido por tres vectores de movimiento; y
derivar, a partir de los valores de los vectores de movimiento del modelo de movimiento afin del
bloque vecino de datos de vídeo, valores de predictores para los vectores de movimiento del modelo
de movimiento afin del bloque actual de datos de vídeo indicado por el segundo elemento de sintaxis;

decodificar, a partir del flujo de bits de vídeo codificado, una representación de las diferencias entre los
valores de los vectores de movimiento del modelo de movimiento afin para el bloque actual de datos de

vídeo y los valores de los predictores;

determinar los valores de los vectores de movimiento del modelo de movimiento afin para el bloque actual de datos de vídeo a partir de los valores de los predictores y las diferencias decodificadas;

determinar, en base a los valores determinados de los vectores de movimiento del modelo de movimiento afin para el bloque actual de datos de vídeo, un bloque predictor de datos de vídeo; y
reconstruir el bloque actual de datos de vídeo en base al bloque predictor de datos de vídeo.

8. El dispositivo de la reivindicación 7, en el que el bloque actual de datos de vídeo se decodifica mediante el uso del modo inter afin y preferentemente en el que el bloque vecino de datos de vídeo se decodifica mediante el uso del modo inter afin o el modo de fusión afin.

9. El dispositivo de la reivindicación 7, en el que el bloque vecino de vídeo comprende un bloque vecino de datos de vídeo seleccionado, y en el que para obtener los valores de los vectores de movimiento del modelo de movimiento afin del bloque vecino de datos de vídeo seleccionado, la una o más unidades de procesamiento se configuran para:

evaluar, en un orden predefinido, bloques vecinos de datos de vídeo del bloque actual de datos de vídeo; y seleccionar un primer bloque vecino de datos de vídeo de la pluralidad de bloques vecinos de datos de vídeo decodificados mediante el uso de compensación de movimiento afin como el bloque vecino de datos de vídeo seleccionado y preferentemente en el que, para decodificar el segundo elemento de sintaxis, la una o más unidades de procesamiento se configuran para decodificar el elemento de sintaxis de una unidad de codificación, CU, que incluye el bloque actual de datos de vídeo.

10. El dispositivo de la reivindicación 9, que comprende además

una pantalla que se configura para mostrar los datos de vídeo reconstruidos; o el dispositivo que comprende uno o más de una cámara, un ordenador, un dispositivo móvil, un dispositivo receptor de difusión o un módulo de conexión.

11. Un dispositivo (20) para codificar un bloque de datos de vídeo, el dispositivo que comprende:

una memoria (64) que se configura para almacenar los datos de vídeo; y una o más unidades de procedimiento que se implementan en circuitos y que se configuran para:

determinar si la compensación de movimiento para un bloque actual de una imagen actual de datos de vídeo se va a realizar mediante el uso de un modelo afin de cuatro parámetros definido por dos vectores de movimiento o un modelo afin de seis parámetros definido por tres vectores de movimiento;

determinar, los valores de los vectores de movimiento del modelo de movimiento afin del bloque actual de datos de vídeo, los vectores de movimiento del modelo de movimiento afin que identifican un bloque predictor de datos de vídeo para el bloque actual de datos de vídeo;

obtener, valores de los vectores de movimiento de un modelo de movimiento afin de un bloque vecino de datos de vídeo;

derivar, a partir de los valores de los vectores de movimiento del modelo de movimiento afin del bloque vecino de datos de vídeo, valores de predictores para los vectores de movimiento del modelo de movimiento afin del bloque actual de datos de vídeo; y

codificar, en un flujo de bits de vídeo codificado, una representación de las diferencias entre los valores de los vectores de movimiento del modelo de movimiento afin para el bloque actual de datos de vídeo y los valores de los predictores;

codificar, en el flujo de bits de vídeo codificado, un primer elemento de sintaxis que se incluye en un conjunto de parámetros de secuencia, SPS, al que hace referencia la imagen actual, en el que un primer valor del primer elemento de sintaxis indica que la compensación de movimiento para las imágenes que hacen referencia al SPS se puede realizar mediante el uso de un modelo de movimiento afin de cuatro parámetros definido por dos vectores de movimiento, y en el que un segundo valor del primer elemento de sintaxis indica que la compensación de movimiento para imágenes que hacen referencia al SPS se puede realizar mediante el uso o bien de un modelo de movimiento afin de cuatro parámetros definido por dos vectores de movimiento o un modelo de movimiento afin de seis parámetros definido por tres vectores de movimiento; y

donde el primer elemento de sintaxis tiene el segundo valor, codificar, en el flujo de bits de vídeo codificado, un segundo elemento de sintaxis en el que un primer valor del segundo elemento de sintaxis indica que la compensación de movimiento para el bloque actual se va a realizar mediante el uso del modelo de movimiento afin de cuatro parámetros definido por dos vectores de movimiento, y en el que un segundo valor del segundo elemento de sintaxis indica que la compensación de movimiento para el bloque actual se va a realizar mediante el uso del modelo de movimiento afin de seis parámetros definido por tres vectores de movimiento.

12. El dispositivo de la reivindicación 9 que comprende además:

un receptor que se configura para recibir los datos de vídeo y almacenar los datos de vídeo en la memoria y preferentemente en el que el dispositivo es un teléfono celular y los datos de vídeo se reciben por el receptor y se modulan de acuerdo con un estándar de comunicación celular.

- 5 13. Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un codificador de vídeo o un decodificador de vídeo ejecuten el procedimiento de una cualquiera de las reivindicaciones 1 a 6.
- 10 14. El procedimiento de la reivindicación 1, en el que, en respuesta al segundo elemento de sintaxis que tiene el segundo valor, determinar cada uno de los valores de los vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo comprende:
determinar los valores de tres vectores de movimiento del modelo de movimiento afín para el bloque actual de datos de vídeo a partir de los valores de los predictores y las diferencias decodificadas.
- 15 15. El procedimiento de la reivindicación 5, en el que el modelo de movimiento afín del bloque actual de datos de vídeo incluye tres vectores de movimiento, y en el que derivar los valores de los predictores para cada uno de los vectores de movimiento del modelo de movimiento afín del bloque actual de datos de vídeo comprende:
derivar, a partir de los valores de los vectores de movimiento del modelo de movimiento afín del bloque vecino de datos de vídeo, valores de los predictores para cada uno de los tres vectores de movimiento del modelo de movimiento afín del bloque actual de datos de vídeo.
- 20

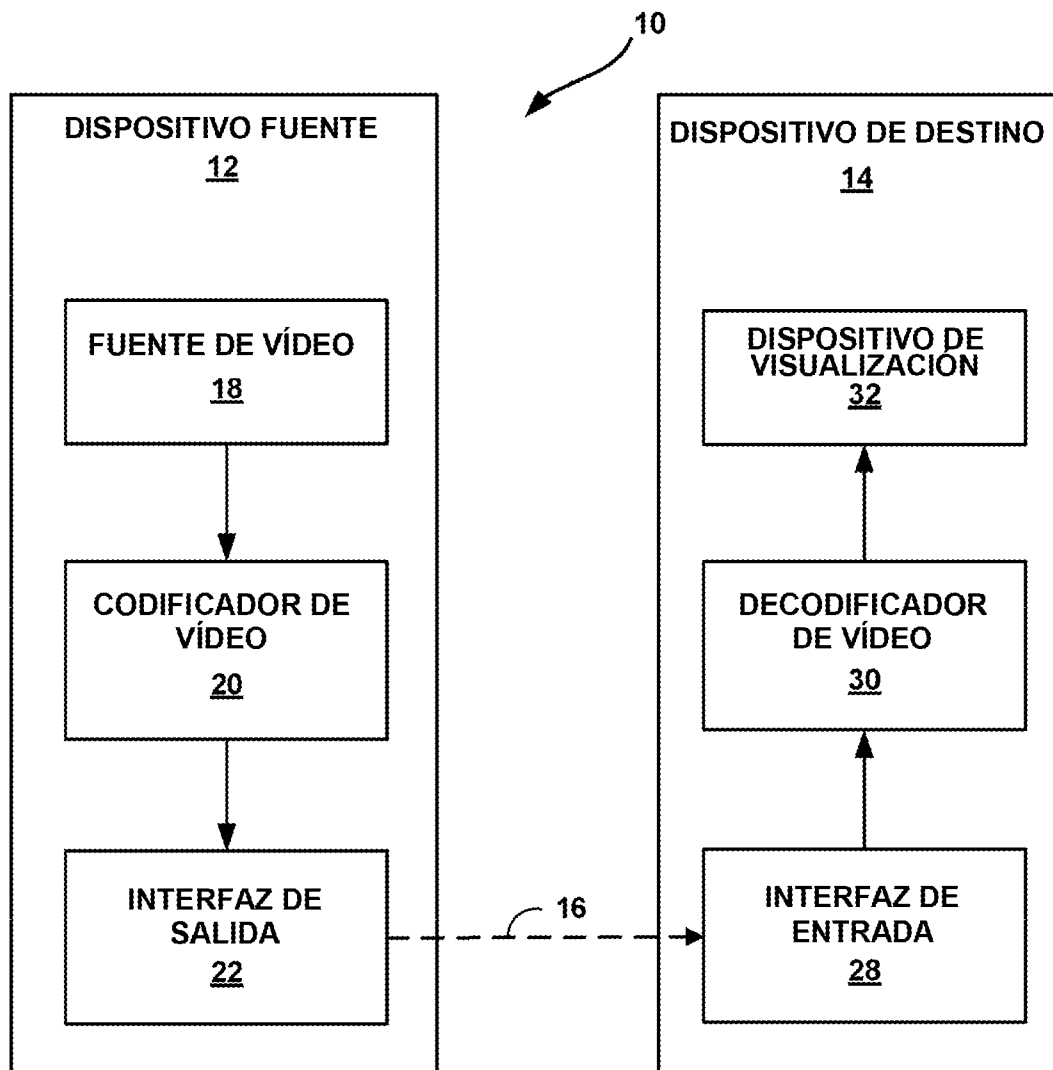


FIGURA 1

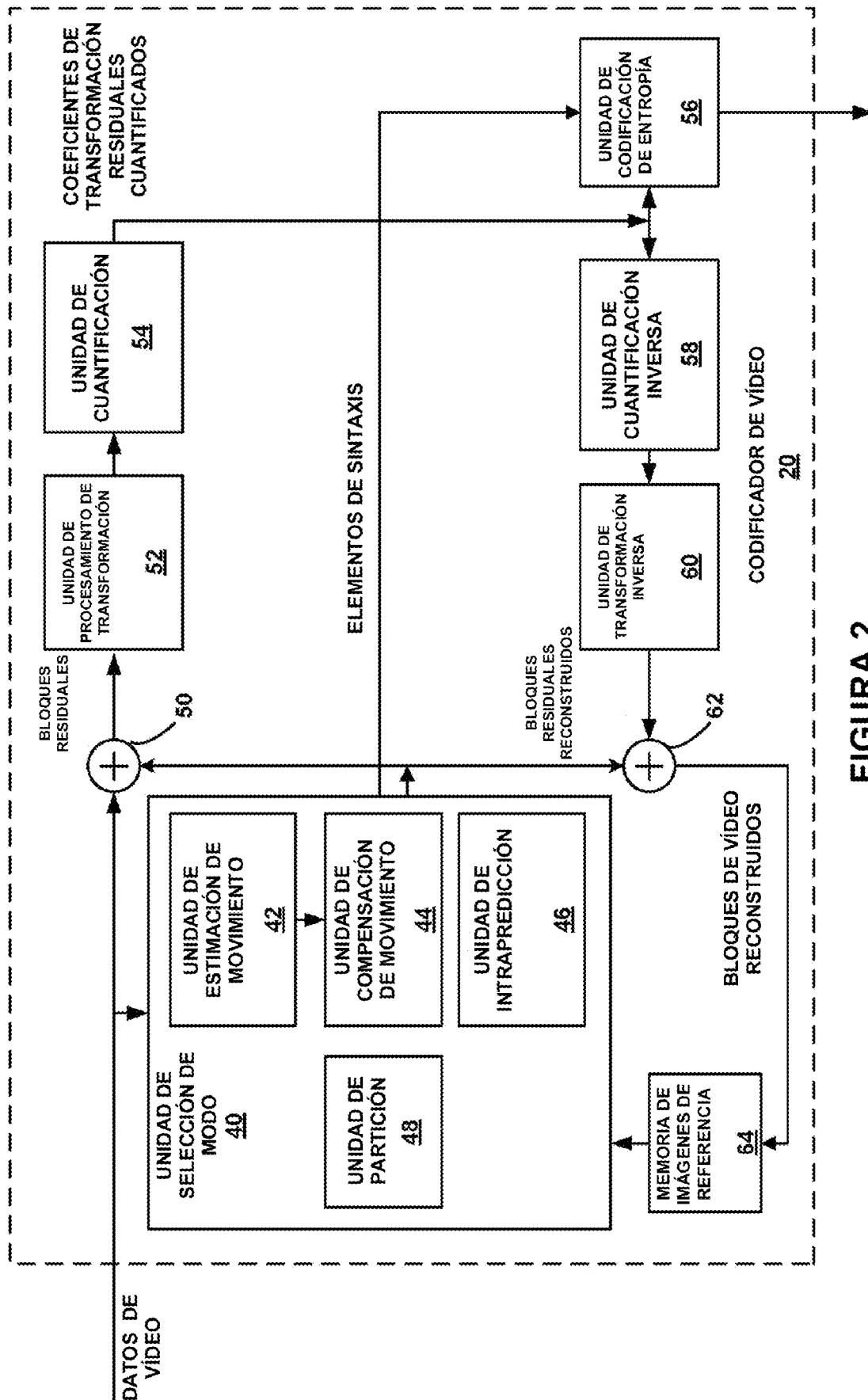


FIGURA 2

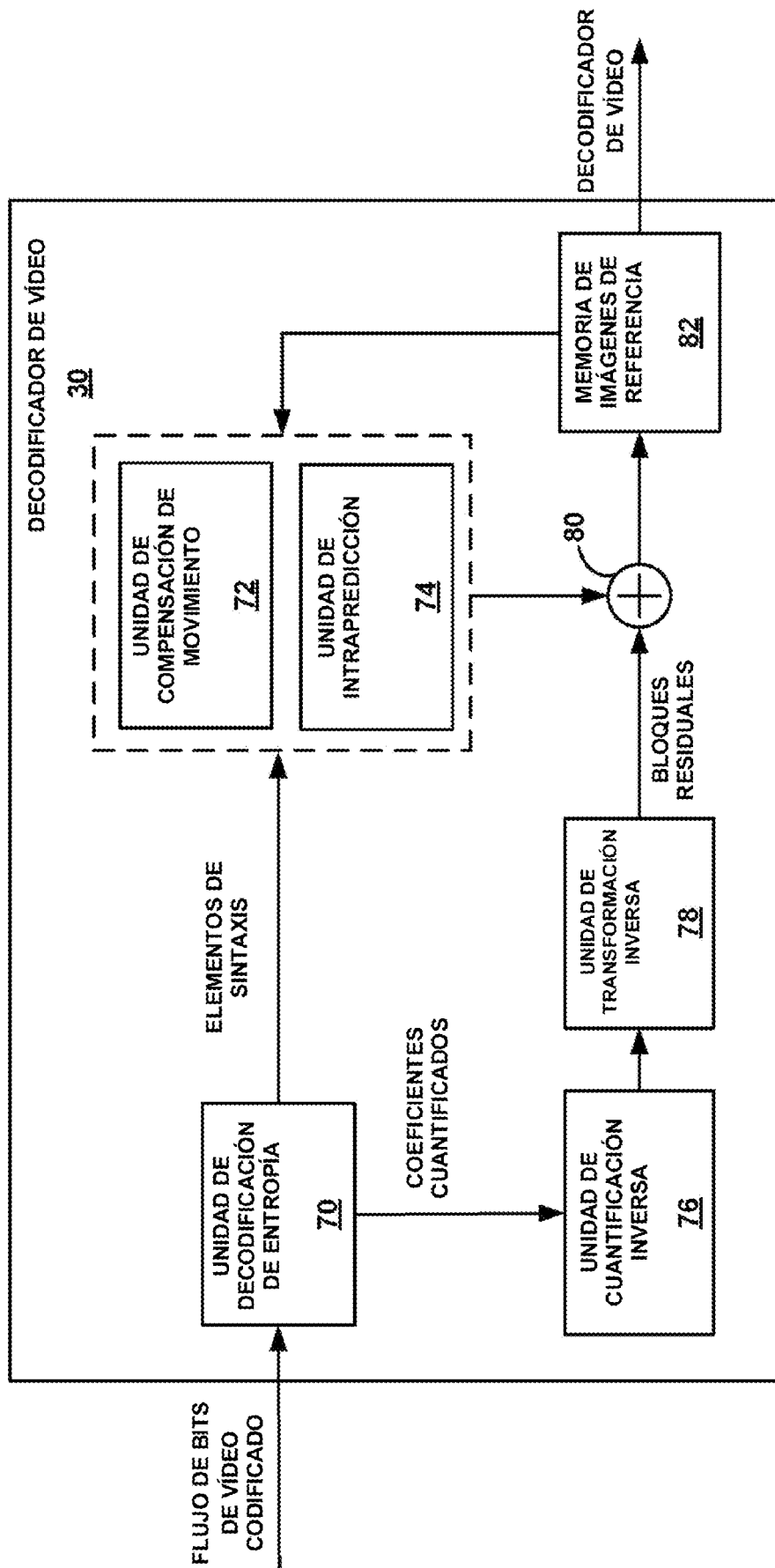
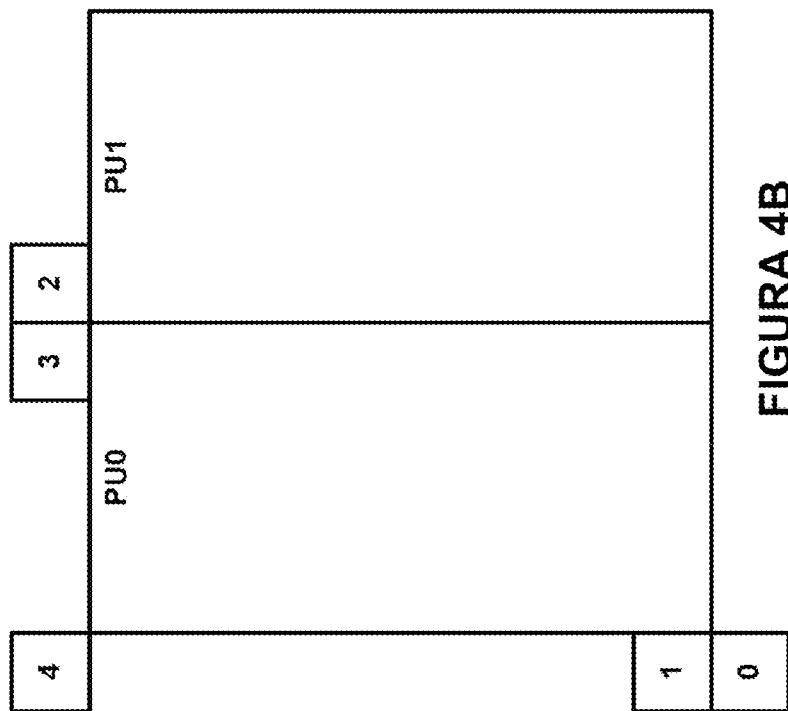
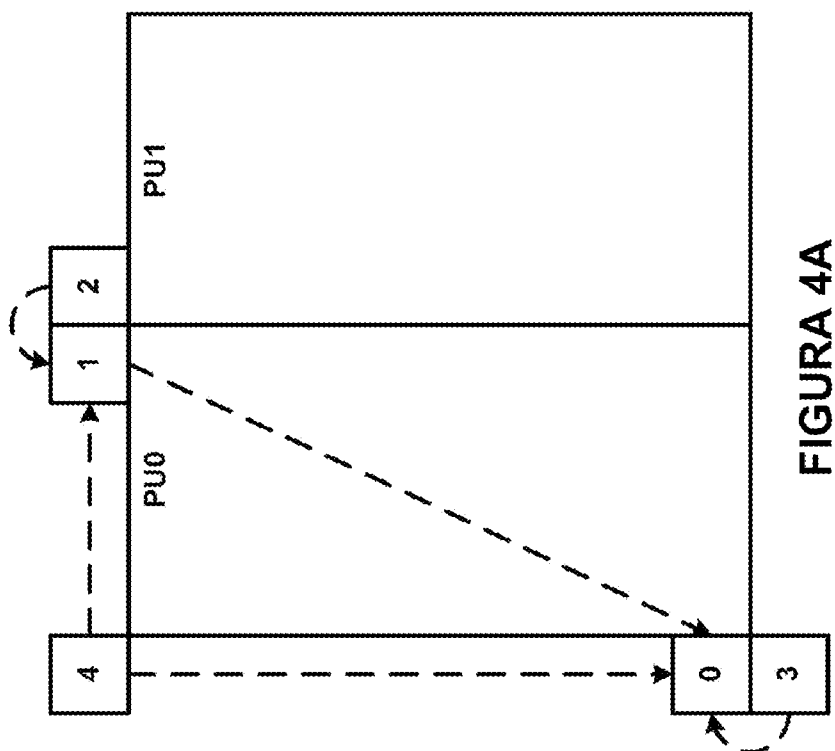


FIGURA 3



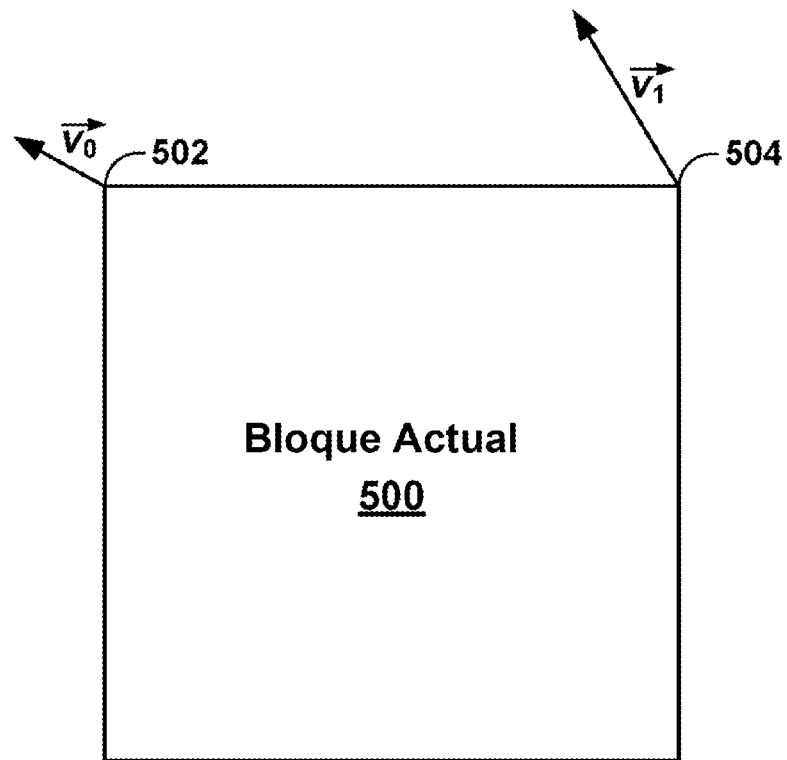


FIGURA 5

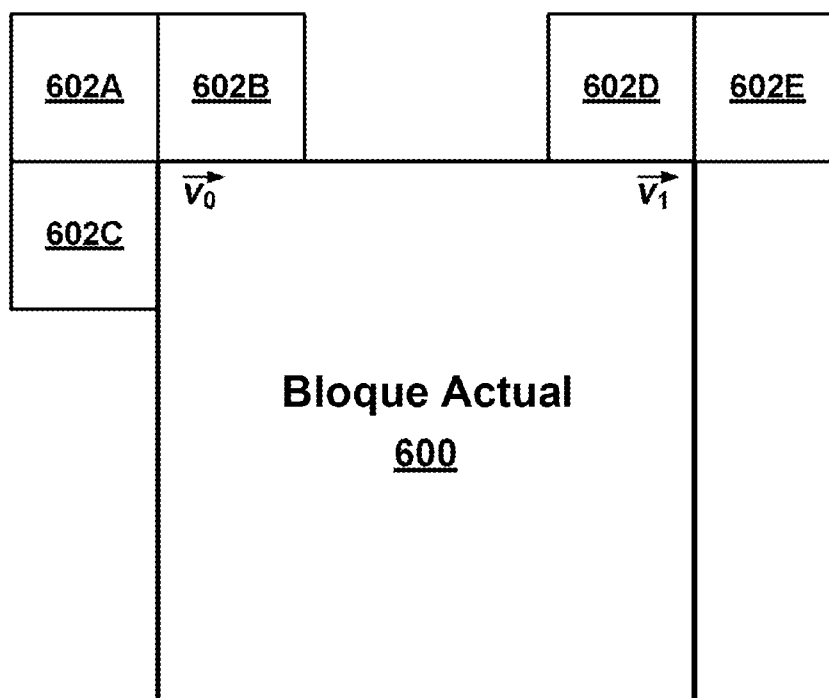


FIGURA 6

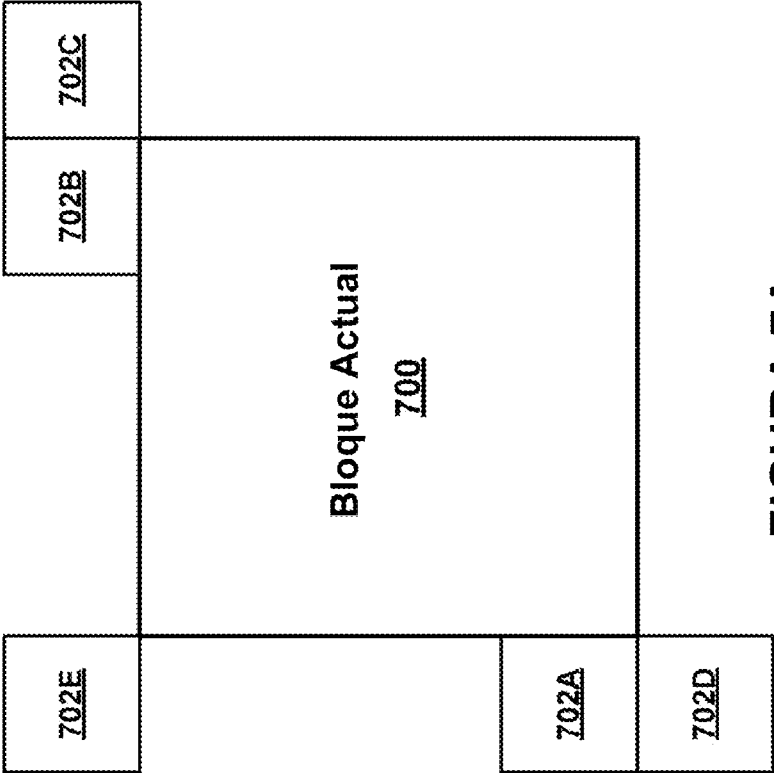


FIGURA 7A

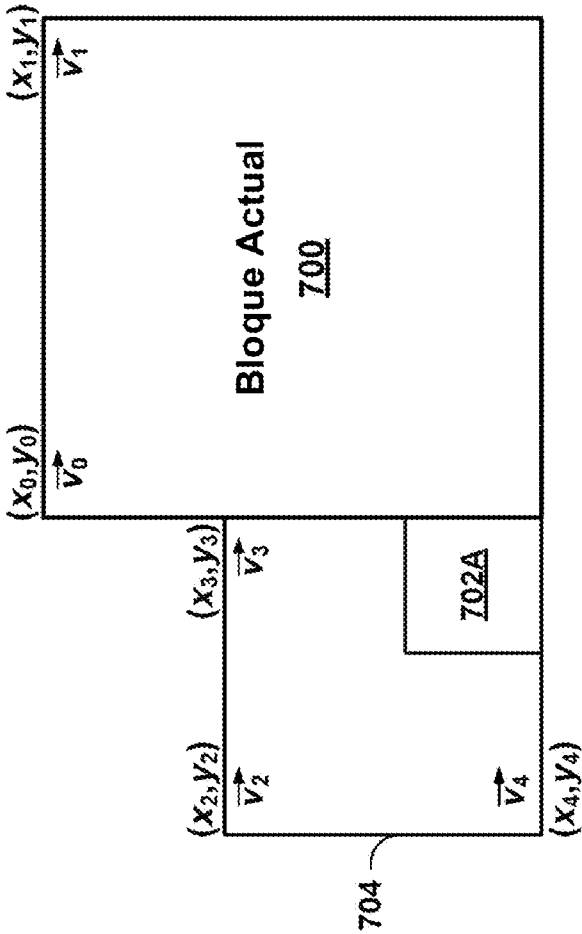


FIGURA 7B

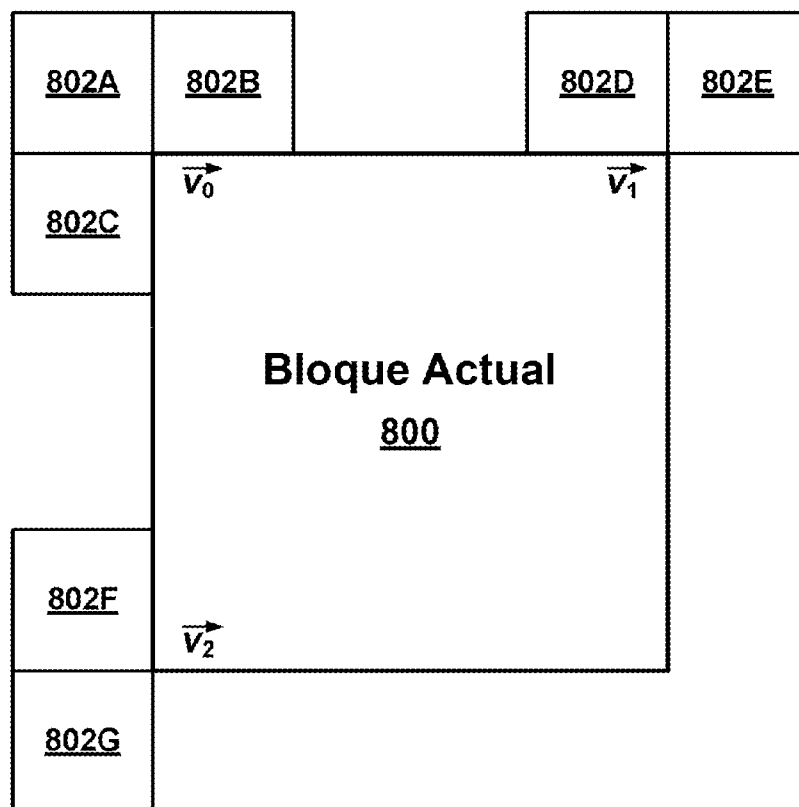


FIGURA 8

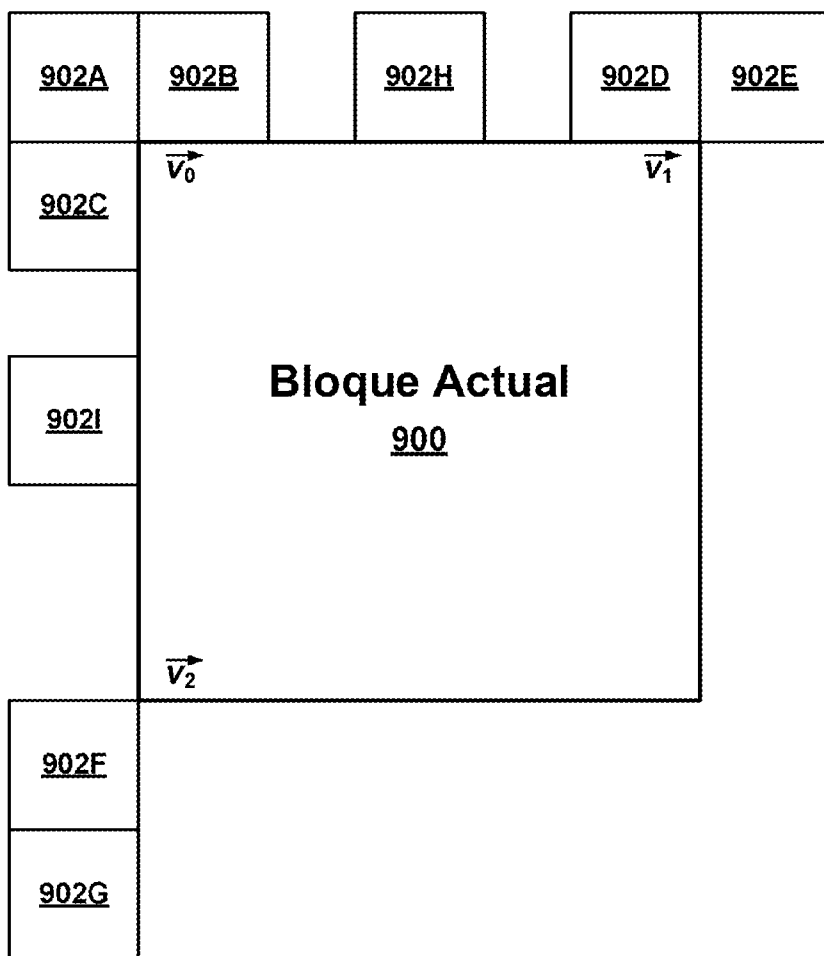


FIGURA 9

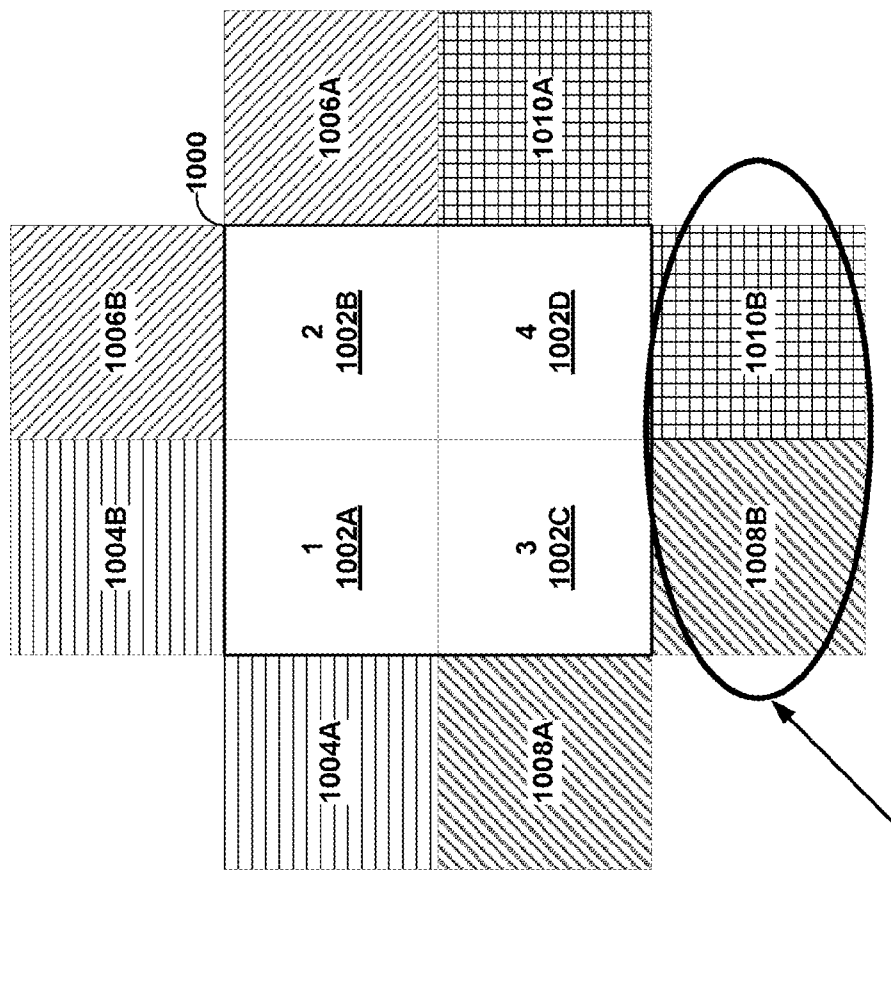


FIGURA 10

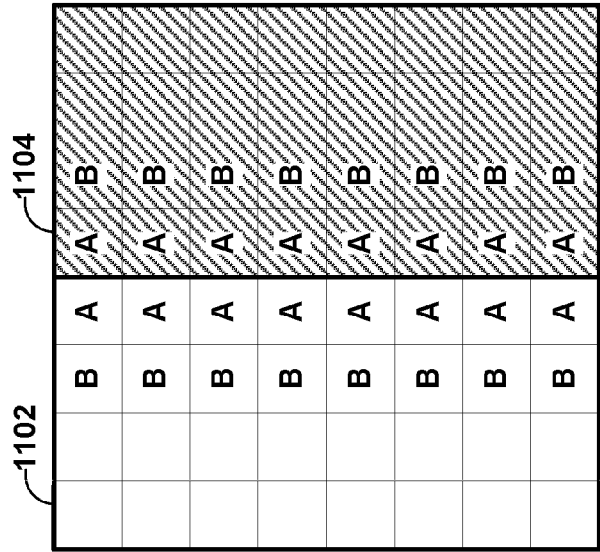


FIGURA 11B

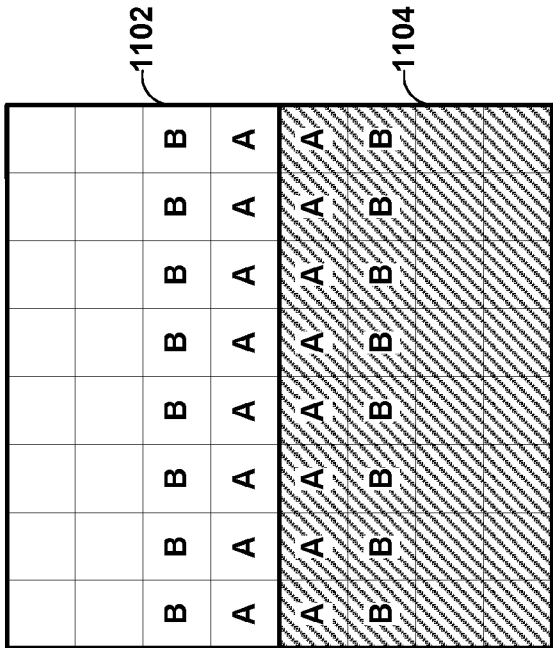


FIGURA 11A

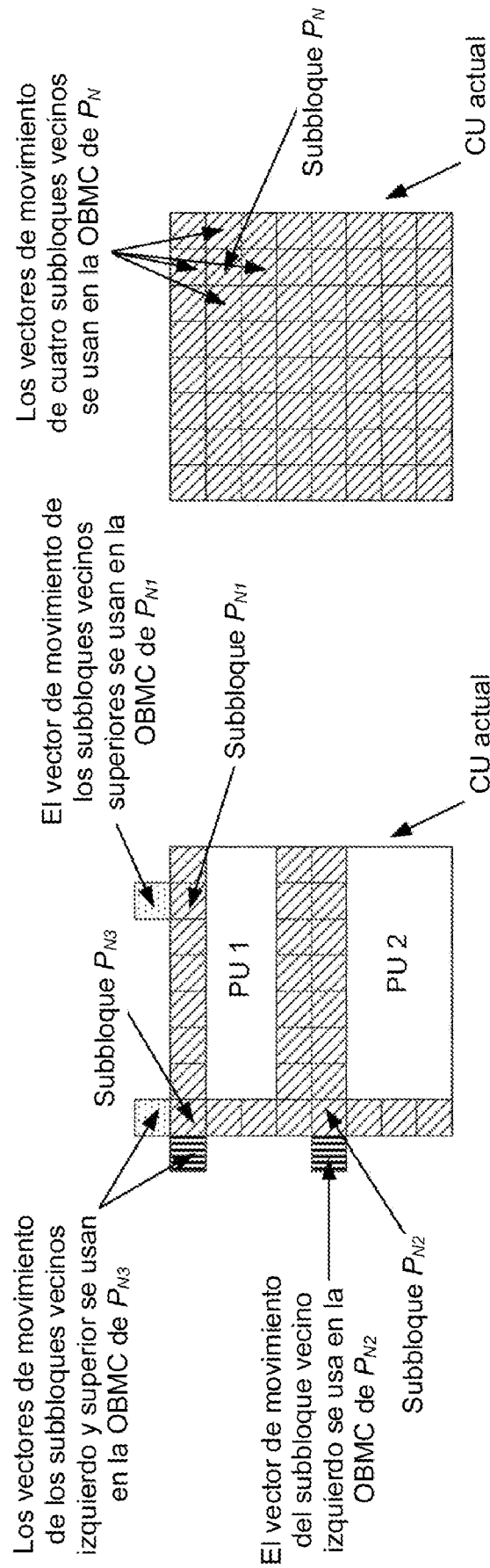


FIGURA 12B

Las Sub-PU en el modo ATMVP

FIGURA 12A

Subbloque en el en el límite de la CU/PU

Subbloque en el que se aplica la OBMC



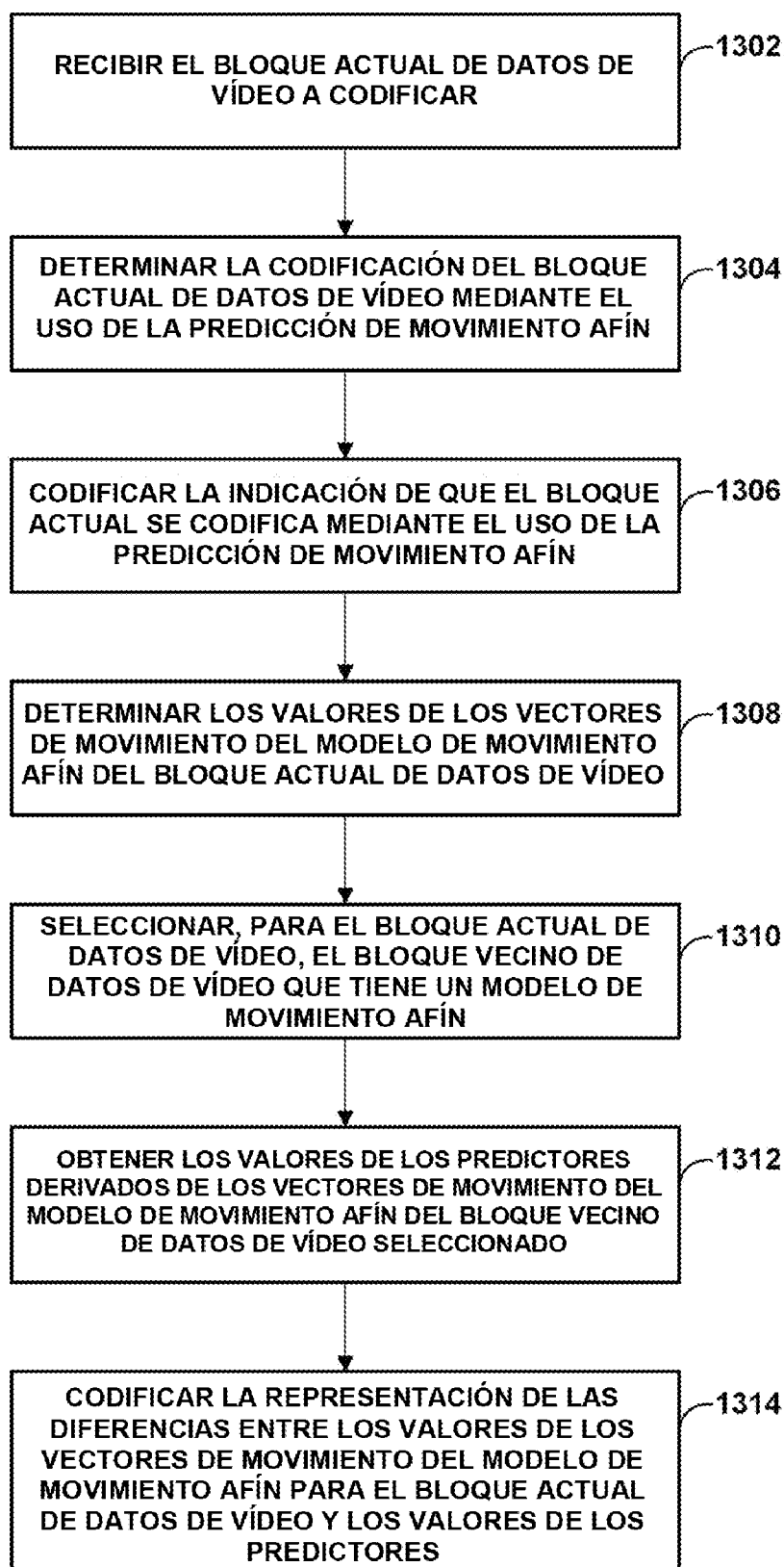


FIGURA 13

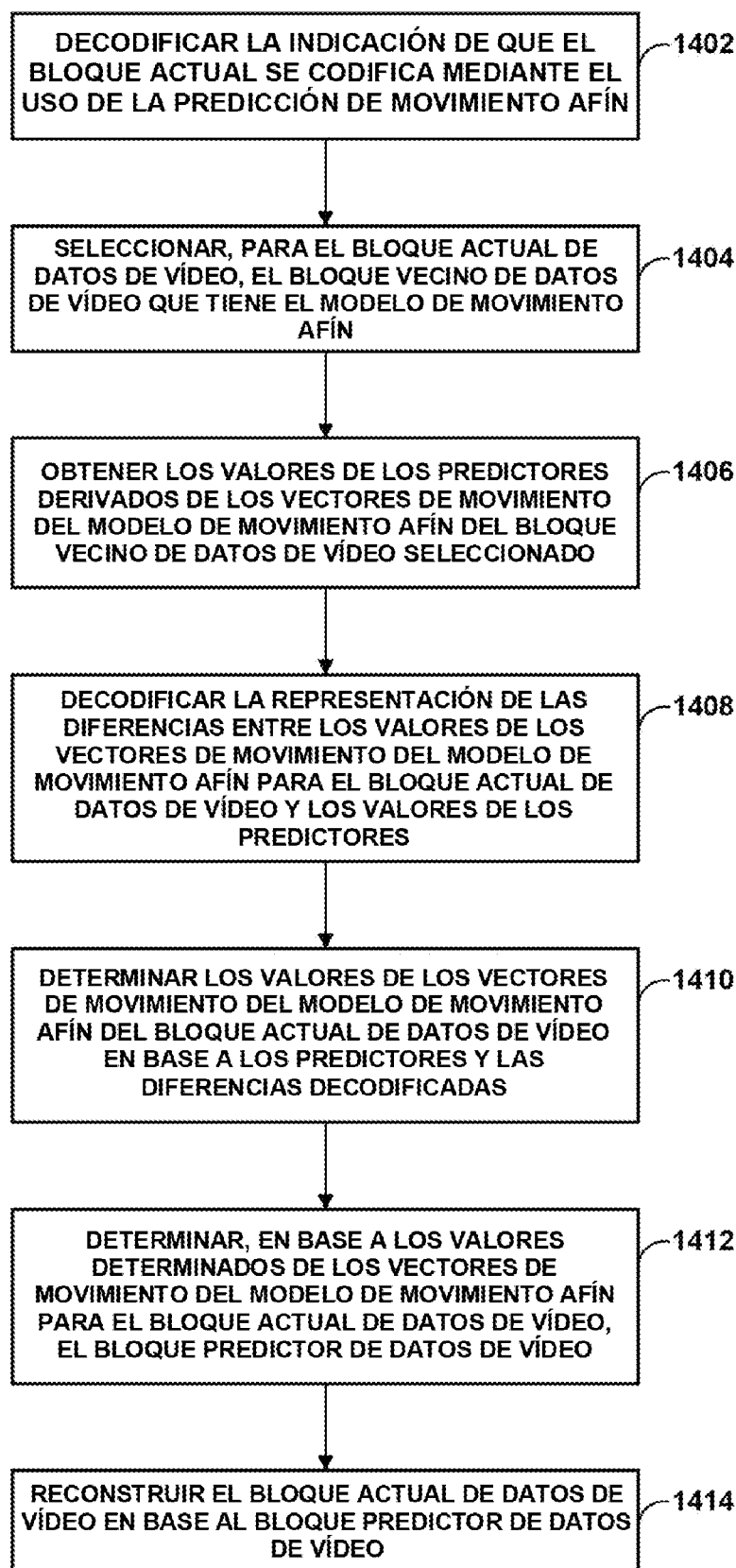


FIGURA 14