

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 982 369**

51 Int. Cl.:

H04N 19/513 (2014.01)
H04N 19/44 (2014.01)
H04N 19/436 (2014.01)
H04N 19/52 (2014.01)
H04N 19/56 (2014.01)
H04N 19/593 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **30.06.2017** **PCT/EP2017/066340**
87 Fecha y número de publicación internacional: **03.01.2019** **WO19001739**
96 Fecha de presentación y número de la solicitud europea: **30.06.2017** **E 17734332 (4)**
97 Fecha y número de publicación de la concesión europea: **08.05.2024** **EP 3635955**

54 Título: **Resiliencia de errores y procesamiento en paralelo para derivación de vector de movimiento de lado de decodificador**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
15.10.2024

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

72 Inventor/es:
ESENLIK, SEMIH;
ZHAO, ZHIJIE y
KOTRA, ANAND, MEHER

74 Agente/Representante:
PONS ARIÑO, Ángel

ES 2 982 369 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

Resiliencia de errores y procesamiento en paralelo para derivación de vector de movimiento de lado de decodificador

5

La presente invención se refiere a derivación de vectores de movimiento en el decodificador y, en particular, a métodos y aparatos para derivación de vector de movimiento que también pueden usarse en codificación y decodificación de imágenes.

10 Antecedentes

Los códecs de video híbridos actuales emplean codificación predictiva. Una imagen de una secuencia de video se subdivide en bloques de píxeles y estos bloques se codifican. En lugar de codificar un bloque píxel por píxel, se predice todo el bloque utilizando píxeles ya codificados en la proximidad espacial o temporal del bloque. El codificador procesa además solo las diferencias entre el bloque y su predicción. El procesamiento adicional típicamente incluye una transformación de los píxeles de bloque en coeficientes en un dominio de transformada. Los coeficientes se pueden comprimir aún más por medio de cuantificación y luego compactarse mediante codificación entrópica para formar un flujo de bits. El flujo de bits incluye además cualquier información de señalización que permite al decodificador decodificar el video codificado. Por ejemplo, la señalización puede incluir ajustes relacionados con los ajustes de codificador, tal como tamaño de la imagen de entrada, velocidad de cuadros, indicación de paso de cuantificación, la predicción aplicada a los bloques de las imágenes, o similares.

La predicción temporal aprovecha la correlación temporal entre imágenes, también conocidas como cuadros, de un video. La predicción temporal también se llama inter-predicción, ya que es una predicción que usa las dependencias entre (inter) diferentes cuadros de video. Por consiguiente, un bloque que se codifica, también denominado bloque actual, se predice a partir de (una o más) imágenes codificadas previamente denominadas imagen de referencia. La imagen de referencia no es necesariamente una imagen que precede a la imagen actual en la que se encuentra el bloque actual en el orden de visualización de la secuencia de video. El codificador puede codificar las imágenes en un orden de codificación diferente del orden de visualización. Como una predicción del bloque actual, se puede determinar un bloque co-ubicado en una imagen de referencia. El bloque co-ubicado es un bloque que se encuentra en la imagen de referencia en la misma posición que el bloque actual en la imagen actual. Esta predicción es precisa para regiones de imágenes inmóviles, es decir, regiones de imágenes sin movimiento de una imagen a otra.

Para obtener un predictor que tenga en cuenta el movimiento, es decir, un predictor de movimiento compensado, se emplea típicamente estimación de movimiento al determinar la predicción del bloque actual. En consecuencia, el bloque actual se predice por un bloque en la imagen de referencia, que se encuentra en una distancia dada por un vector de movimiento de la posición del bloque co-ubicado. Para permitir que un decodificador determine la misma predicción del bloque actual, el vector de movimiento puede señalizarse en el flujo de bits. Para reducir aún más la sobrecarga de señalización producida por la señalización del vector de movimiento para cada uno de los bloques, se puede estimar el vector de movimiento en sí. La estimación del vector de movimiento puede realizarse basándose en los vectores de movimiento de los bloques vecinos en el dominio espacial y/o temporal.

La predicción del bloque actual se puede calcular utilizando una imagen de referencia o ponderando las predicciones obtenidas de dos o más imágenes de referencia. La imagen de referencia puede ser una imagen adyacente, es decir, una imagen inmediatamente anterior y/o la imagen inmediatamente posterior a la imagen actual en el orden de visualización, ya que las imágenes adyacentes tienen más probabilidad de ser similares a la imagen actual. Sin embargo, en general, la imagen de referencia también puede ser cualquier otra imagen que preceda o siga a la imagen actual en el orden de visualización y que preceda a la imagen actual en el flujo de bits (orden de decodificación). Esto puede proporcionar ventajas, por ejemplo, en caso de oclusiones y/o movimientos no lineales en el contenido de video. La imagen de referencia también se puede señalar en el flujo de bits.

Un modo especial de la inter-predicción es la llamada bi-predicción en la que se utilizan dos imágenes de referencia para generar la predicción del bloque actual. En particular, dos predicciones determinadas en las dos imágenes de referencia respectivas se combinan en una señal de predicción del bloque actual. La bi-predicción puede dar como resultado una predicción más precisa del bloque actual que la uni-predicción, es decir, predicción que solo usa una única imagen de referencia. La predicción más precisa conduce a diferencias más pequeñas entre los píxeles de bloque actual y la predicción (llamadas también como "residuales"), que pueden codificarse de manera más eficiente, es decir, comprimidas en un flujo de bits más corto.

Para proporcionar una estimación de movimiento más precisa, la resolución de la imagen de referencia puede mejorarse interpolando muestras entre píxeles. La interpolación fraccional de píxeles se puede realizar mediante un promedio ponderado de los píxeles más cercanos. En el caso de resolución de medio píxel, por

ejemplo, se usa típicamente una interpolación bilineal. Otros píxeles fraccionarios se calculan como un promedio de los píxeles más cercanos ponderados por la inversa de la distancia entre los píxeles más cercanos respectivos al píxel que se predice.

- 5 La estimación del vector de movimiento es una tarea computacionalmente compleja en la que se calcula una similitud entre el bloque actual y los bloques de predicción correspondientes señalados por los vectores de movimiento candidatos en la imagen de referencia. Para reducir la complejidad, el número de vectores de movimiento candidatos generalmente se reduce al limitar los vectores de movimiento candidatos a un cierto espacio de búsqueda. El espacio de búsqueda puede estar, por ejemplo, definido por un número y/o posiciones de píxeles que rodean la posición en la imagen de referencia correspondiente a la posición del bloque actual en la imagen actual. Por otro lado, los vectores de movimiento candidatos pueden definirse mediante una lista de vectores de movimiento candidatos formados por vectores de movimiento de bloques vecinos.

- 15 Generalmente los vectores de movimiento se determinan al menos parcialmente en el lado de codificador y se señalan al decodificador dentro del flujo de bits codificado. Sin embargo, los vectores de movimiento también pueden derivarse en el decodificador. En este caso, el bloque actual no está disponible en el decodificador y no puede usarse para calcular la similitud con los bloques a los que apuntan los vectores de movimiento candidatos en la imagen de referencia. Por lo tanto, en lugar del bloque actual, se utiliza una plantilla que se construye a partir de píxeles de bloques ya decodificados. Por ejemplo, se pueden usar píxeles ya decodificados adyacentes al bloque actual. Esta estimación de movimiento proporciona una ventaja de reducir la señalización: el vector de movimiento se deriva de la misma manera tanto en el codificador como en el decodificador y, por lo tanto, no se necesita señalización. Por otro lado, puede ser menor la precisión de tal estimación de movimiento.

- 25 Para proporcionar una compensación entre precisión y sobrecarga de señalización, la estimación del vector de movimiento se puede dividir en dos pasos: derivación de vector de movimiento y refinamiento de vector de movimiento. Por ejemplo, una derivación de vector de movimiento puede incluir selección de un vector de movimiento de la lista de candidatos. Este vector de movimiento seleccionado puede refinarse adicionalmente, por ejemplo, mediante una búsqueda dentro de un espacio de búsqueda. La búsqueda en el espacio de búsqueda se basa en el cálculo de la función de costo para cada vector de movimiento candidato, es decir, para cada posición candidata del bloque al que apunta el vector de movimiento candidato.

- El documento JVET-D0029: Decoder-Side Motion Vector Refinement Based on Bilateral Template Matching, X. Chen, J. An, J. Zheng (El documento se puede encontrar en: <http://phenix.it-sudparis.eu/ivet/> site) muestra el refinamiento de vector de movimiento en el que se encuentra un primer vector de movimiento en resolución de píxeles enteros y se refina aún más mediante una búsqueda con una resolución de medio píxel en un espacio de búsqueda alrededor del primer vector de movimiento. El documento JVET-B0047: TE1: Fast techniques to improve self derivation of motion estimation, Y-JEN CHIU ET AL, 2. JCT-VC MEETING; 21-7-2010 - 28-7-2010; GINEBRA; (EQUIPO CONJUNTO DE COLABORACIÓN SOBRE CODIFICACIÓN DE VIDEO DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); 28 de julio de 2010 propone un proceso de predicción de vector de movimiento amigable en paralelo para la generación de listas de predicción de vector de movimiento usando únicamente vectores de movimiento vecinos temporales pero no espaciales derivados por la coincidencia de plantillas.

45 Compendio

- Partiendo de los enfoques descritos anteriormente, el objetivo de la presente descripción es proporcionar una posibilidad mejorada de decodificación paralela de unidades de procesamiento y/o aumentar la resiliencia de errores de la decodificación. Este problema se resuelve mediante el objeto de las reivindicaciones independientes. En las reivindicaciones dependientes se proporcionan otras formas de implementación.

- Para lograr esto, los vectores de movimiento de vecinos a un bloque actual que se han derivado mediante el uso de la coincidencia de plantillas no se utilizan para predecir el vector de movimiento del bloque actual, o, alternativamente, solo se omite la verificación de redundancia del vector de movimiento para estos vectores de movimiento durante la generación de la lista de predicción de vector de movimiento.

- En otras palabras, los vectores de movimiento de los vecinos que se han derivado usando la imagen de referencia no se usan o se usan sin verificación de redundancia para predecir el vector de movimiento del bloque actual.

- Esto puede ser derivación y reemplazo usando coincidencia de plantillas. En las realizaciones a continuación, se hace referencia a la aplicación de "coincidencia de plantillas". Sin embargo, en general, también puede considerarse cualquier enfoque que use una imagen de referencia.

- En un primer aspecto, se proporciona un decodificador para determinar una predicción de un bloque actual basándose en una predicción de vector de movimiento del bloque actual, estando el decodificador configurado

para:

- 5 **a.)** recibir señalización de control de un codificador, especificando la señalización de control si el decodificador debe aplicar un primer modo para determinar la predicción de vector de movimiento del bloque actual o si el decodificador debe aplicar un segundo modo para determinar la predicción de vector de movimiento del bloque actual;
- 10 **b.)** determinar, basándose en la señalización de control, si el decodificador debe aplicar el primer modo o el segundo modo para determinar la predicción de vector de movimiento del bloque actual;
- 10 **c.)** en caso de que se determine en el paso **b.)** que el decodificador debe aplicar el primer modo para determinar la predicción de vector de movimiento del bloque actual:
 - 15 **c1.)** detectar, para cada uno de los múltiples bloques adyacentes al bloque actual, si se ha determinado o no un vector de movimiento del bloque adyacente al bloque actual por coincidencia de plantillas,
o si el vector de movimiento del bloque adyacente al bloque actual no se ha determinado por coincidencia de plantillas, incluir el vector de movimiento del bloque adyacente como un vector de movimiento candidato para la predicción de vector de movimiento del bloque actual en una primera lista,
20 o si el vector de movimiento del bloque adyacente al bloque actual se ha determinado por coincidencia de plantillas, no incluir el vector de movimiento del bloque adyacente en la primera lista, sino incluir un vector de movimiento de reemplazo como un vector de movimiento candidato para la predicción de vector de movimiento del bloque actual en la primera lista,
25 **c2.)** seleccionar un primer vector de movimiento de la primera lista como una estimación de la predicción de vector de movimiento del bloque actual, determinar la predicción de vector de movimiento del bloque actual de acuerdo con la estimación y realizar la predicción del bloque actual basándose en la predicción de vector de movimiento seleccionado del bloque actual;
30 **d.)** en caso de que se determine en el paso **b.)** que el decodificador debe aplicar el segundo modo para determinar la predicción de vector de movimiento del bloque actual:
 - 35 **d1.)** detectar, para cada uno de los múltiples bloques adyacentes al bloque actual, si se ha determinado o no el vector de movimiento del bloque adyacente al bloque actual por coincidencia de plantillas,
40 **d11.)** si el vector de movimiento del bloque adyacente al bloque actual no se ha determinado por coincidencia de plantillas, verificar si el vector de movimiento del bloque adyacente al bloque actual ya está incluido en una segunda lista,
o si el vector de movimiento del bloque adyacente al bloque actual ya está incluido en la segunda lista, no incluir el vector de movimiento del bloque adyacente al bloque actual en la segunda lista;
45 o si el vector de movimiento del bloque adyacente al bloque actual no está incluido en la segunda lista, incluir el vector de movimiento del bloque adyacente al bloque actual en la segunda lista;
50 **d12.)** si el vector de movimiento del bloque adyacente al bloque actual se ha determinado por coincidencia de plantillas, no realizar la verificación, sino agregar el vector de movimiento del bloque adyacente al bloque actual a la segunda lista;
55 **d2.)** seleccionar un segundo vector de movimiento de la segunda lista como una estimación de la predicción de vector de movimiento del bloque actual, determinar la predicción de vector de movimiento del bloque actual de acuerdo con la estimación, y realizar la predicción del bloque actual basándose en la predicción de vector de movimiento seleccionado del bloque actual.
60 En una primera forma de implementación del primer aspecto, el vector de movimiento de reemplazo es un vector de movimiento de un bloque vecino al bloque actual y diferente del bloque adyacente.
60 En un segundo aspecto se proporciona un método para determinar una predicción de un bloque actual basándose en una predicción de vector de movimiento del bloque actual realizada por un decodificador, comprendiendo el método los pasos de:
 - 65 **a.)** recibir señalización de control de un codificador, especificando la señalización de control si el decodificador debe aplicar un primer modo para determinar la predicción de vector de movimiento del bloque actual o si el decodificador debe aplicar un segundo modo para determinar la predicción de vector de movimiento del bloque actual;

- b.)** determinar, basándose en la señalización de control, si el decodificador debe aplicar el primer modo o el segundo modo para determinar la predicción de vector de movimiento del bloque actual;
- 5 **c.)** en caso de que se determine en el paso **b.)** que el decodificador debe aplicar el primer modo para determinar la predicción de vector de movimiento del bloque actual:
- 10 c1.) detectar, para cada uno de los múltiples bloques adyacentes al bloque actual, si se ha determinado o no un vector de movimiento del bloque adyacente al bloque actual por coincidencia de plantillas,
- o si el vector de movimiento del bloque adyacente al bloque actual no se ha determinado por coincidencia de plantillas, incluir el vector de movimiento del bloque adyacente como un vector de movimiento candidato para la predicción de vector de movimiento del bloque actual en una primera lista,
- 15 o si el vector de movimiento del bloque adyacente al bloque actual se ha determinado por coincidencia de plantillas, no incluir el vector de movimiento del bloque adyacente en la primera lista, sino incluir un vector de movimiento de reemplazo como un vector de movimiento candidato para la predicción de vector de movimiento del bloque actual en la primera lista,
- 20 **c2.)** seleccionar un primer vector de movimiento de la primera lista como una estimación de la predicción de vector de movimiento del bloque actual, y determinar la predicción de vector de movimiento del bloque actual de acuerdo con la estimación y realizar la predicción del bloque actual basándose en la predicción de vector de movimiento seleccionado del bloque actual;
- 25 **d.)** en caso de que se determine en el paso **b.)** que el decodificador debe aplicar el segundo modo para determinar la predicción de vector de movimiento del bloque actual:
- 30 **d1.)** detectar, para cada uno de los múltiples bloques adyacentes al bloque actual, si se ha determinado o no el vector de movimiento del bloque adyacente al bloque actual por coincidencia de plantillas,
- d11.)** si el vector de movimiento del bloque adyacente al bloque actual no se ha determinado por coincidencia de plantillas, verificar si el vector de movimiento del bloque adyacente al bloque actual ya está incluido en una segunda lista,
- 35 o si el vector de movimiento del bloque adyacente al bloque actual ya está incluido en la segunda lista, no incluir el vector de movimiento del bloque adyacente al bloque actual en la segunda lista;
- o si el vector de movimiento del bloque adyacente al bloque actual no está incluido en la segunda lista, incluir el vector de movimiento del bloque adyacente al bloque actual en la segunda lista;
- 40 **d12.)** si el vector de movimiento del bloque adyacente al bloque actual se ha determinado por la coincidencia de plantillas, no realizar la verificación, sino agregar el vector de movimiento del bloque adyacente al bloque actual a la segunda lista;
- 45 **d2.)** seleccionar un segundo vector de movimiento de la segunda lista como una estimación de la predicción de vector de movimiento del bloque actual, determinar la predicción de vector de movimiento del bloque actual de acuerdo con la estimación, y realizar la predicción del bloque actual basándose en la predicción de vector de movimiento seleccionado del bloque actual.
- 50 Breve descripción de las figuras
- En lo siguiente se describen con más detalle realizaciones de ejemplo con referencia a las figuras y dibujos adjuntos, en los que:
- 55 la figura 1 es un diagrama de bloques que muestra una estructura de ejemplo de un codificador de video que es un ejemplo que no forma parte de la invención.
- La figura 2 es un diagrama de bloques que muestra una estructura de ejemplo de un decodificador de video.
- 60 La figura 3 es un diagrama de bloques que muestra una estructura de ejemplo de un aparato para determinar una predicción para un bloque de imagen.
- La figura 4 es un dibujo esquemático de un bloque actual y bloques vecinos de ejemplo que se pueden usar para generación de listas de candidatos.
- 65 La figura 5 es un dibujo esquemático que ilustra procesamiento de decodificador y posibilidad de paralelización

si se aplica refinamiento de vector de movimiento.

La figura 6 es un diagrama de bloques que ilustra la unidad de inter-predicción de acuerdo con una realización.

- 5 La figura 7 es un dibujo esquemático que ilustra procesamiento de decodificador y posibilidad de paralelización si se aplica refinamiento de vector de movimiento.

- 10 La figura 8 es un diagrama de flujo que ilustra un método para restringir la determinación del vector de movimiento con base en la determinación del vector de movimiento de los bloques vecinos de acuerdo con un ejemplo que no forma parte de la invención.

La figura 9 es un dibujo esquemático que ilustra un bloque actual de ejemplo y sus vecinos.

- 15 La figura 10 es un dibujo esquemático que ilustra un bloque actual de ejemplo y su vecino con un bloque adyacente al vecino.

La figura 11 es un dibujo esquemático que ilustra el procesamiento de decodificador y se aplica posibilidad de omisión de verificación de paralelización.

- 20 La figura 12 es un dibujo esquemático que ilustra un bloque actual de ejemplo y sus vecinos para predecir el vector de movimiento del bloque actual de acuerdo con un ejemplo que no forma parte de la invención.

La figura 13 es un dibujo esquemático que ilustra un bloque actual de ejemplo y sus vecinos para intra-predicción del bloque actual de acuerdo con un ejemplo que no forma parte de la invención.

- 25 La figura 14 es un dibujo esquemático que ilustra una región de ejemplo para limitar la determinación del vector de movimiento de un bloque actual.

- 30 La figura 15 es un dibujo esquemático que ilustra una región de ejemplo para limitar la determinación del vector de movimiento de un bloque actual.

La figura 16 es un dibujo esquemático que ilustra los errores que pueden ocurrir en recepción de una secuencia de video.

- 35 La figura 17 es un dibujo esquemático que ilustra errores que pueden ocurrir en recepción de una secuencia de video.

Descripción detallada de las realizaciones de la invención y ejemplos que no forman parte de la invención

- 40 La presente divulgación se refiere a determinación de una predicción para un bloque actual y, en particular, a predicción cruzada. Puede emplearse en la estimación de movimiento aplicada durante codificación y decodificación de video. A continuación, se describen a continuación codificador y decodificador de ejemplo que pueden implementar la estimación de movimiento empleando la construcción de espacio de búsqueda de la presente divulgación.

- 45 La figura 1 muestra un codificador 100 que es un ejemplo que no forma parte de la invención que comprende una entrada para recibir bloques de entrada de cuadros o imágenes de un flujo de video y una salida para generar un flujo de bits de video codificado. El término "cuadro" en esta divulgación se usa como sinónimo de imagen. Sin embargo, se observa que la presente divulgación también es aplicable a los campos en caso de que se aplique entrelazado. En general, una imagen incluye m veces n píxeles. Esto corresponde a muestras de imágenes y puede comprender uno o más componentes de color. Por simplicidad, la siguiente descripción se refiere a píxeles que significan muestras de luminancia. Sin embargo, se observa que la búsqueda de vectores de movimiento de la invención se puede aplicar a cualquier componente de color, que incluye crominancia o componentes de un espacio de búsqueda tal como RGB o similares. Por otro lado, puede ser
50 beneficioso realizar solo la estimación del vector de movimiento para un componente y aplicar el vector de movimiento determinado a más (o todos) componentes.
55

Los bloques de entrada que se van codificar no tienen necesariamente el mismo tamaño. Una imagen puede incluir bloques de diferentes tamaños y la cuadrícula de bloques de diferentes imágenes también puede diferir.

- 60 En una realización explicativa, el codificador 100 se adapta para aplicar predicción, transformación, cuantificación y codificación entrópica al flujo de video. La transformación, cuantificación y codificación entrópica se llevan a cabo respectivamente por una unidad de transformación 101, una unidad de cuantificación 102 y una unidad de codificación entrópica 103 para generar como salida el flujo de bits de video codificado.

- 65 El flujo de video puede incluir una pluralidad de cuadros, en donde cada cuadro se divide en bloques de cierto

tamaño que se intra- o inter-codifican. Los bloques de, por ejemplo, el primer cuadro de la secuencia de video se codifican por medio de una unidad de intra- predicción 109. Un intra-cuadro se codifica utilizando solo la información dentro del mismo cuadro, de modo que puede decodificarse independientemente y puede proporcionar un punto de entrada en el flujo de bits para acceso aleatorio. Los bloques de otros cuadros del flujo de video se pueden inter-codificar por medio de una unidad de inter-predicción 110: la información de cuadros previamente codificados (cuadros de referencia) se usa para reducir la redundancia temporal, de modo que se predice cada bloque de un cuadro inter-codificado de un bloque en un cuadro de referencia. Una unidad de selección de modo 108 se adapta para seleccionar si un bloque de un cuadro se debe procesar por la unidad de intra-predicción 109 o la unidad de inter-predicción 110. Este bloque también controla los parámetros de intra- o inter-predicción. Para permitir la actualización de la información de la imagen, se puede proporcionar un bloque intra-codificado dentro de cuadros inter-codificados. Además, los intra- cuadros que contienen solo bloques intra-codificados pueden insertarse regularmente en la secuencia de video para proporcionar puntos de entrada para decodificación, es decir, puntos donde el decodificador puede comenzar a decodificar sin tener información de las tramas previamente codificadas.

La unidad de intra-predicción 109 es una unidad de predicción de bloque. Para realizar predicción espacial o temporal, los bloques codificados pueden procesarse adicionalmente por una unidad de cuantificación inversa 104 y una unidad de transformación inversa 105. Después de la reconstrucción del bloque, se aplica una unidad de filtración de bucle 106 para mejorar aún más la calidad de la imagen decodificada. Los bloques filtrados luego forman los cuadros de referencia que luego se almacenan en una memoria intermedia de cuadros 107. Este bucle de decodificación (decodificador) en el lado de codificador proporciona la ventaja de producir cuadros de referencia que son iguales a las imágenes de referencia reconstruidas en el lado de decodificador. En consecuencia, el lado de codificador y el decodificador funcionan de manera correspondiente. El término "reconstrucción" se refiere aquí a obtener el bloque reconstruido agregando al bloque residual decodificado el bloque de predicción.

La unidad de inter-predicción 110 recibe como entrada un bloque de un cuadro o imagen actual para que se codifique y uno o varios cuadros de referencia o imágenes de la memoria intermedia de cuadros 107. La unidad de predicción 110 aplica la estimación de movimiento y la compensación de movimiento. La estimación de movimiento se utiliza para obtener un vector de movimiento y un cuadro de referencia con base en cierta función de costo. La compensación de movimiento luego describe un bloque actual del cuadro actual en términos de la traslación de un bloque de referencia del cuadro de referencia al cuadro actual, es decir, mediante un vector de movimiento. La unidad de inter-predicción 110 emite el bloque de predicción para el bloque actual, en donde este bloque de predicción minimiza la función de costo. Por ejemplo, la función de costo puede ser una diferencia entre el bloque actual que se va a codificar y su bloque de predicción, es decir, la función de costo minimiza el bloque residual. La minimización del bloque residual se basa por ejemplo en el cálculo de una suma de diferencias absolutas (SAD) entre todos los píxeles (muestras) del bloque actual y el bloque candidato en la imagen de referencia candidata. Sin embargo, en general, se puede emplear cualquier otra métrica de similitud, como error cuadrático medio (MSE) o métrica de similitud estructural (SSIM).

Sin embargo, la función de costo también puede ser el número de bits necesarios para codificar esta interbloque y/o distorsión resultante de esta codificación. Por lo tanto, el procedimiento de optimización de la distorsión de la velocidad se puede usar para decidir sobre la selección de vector de movimiento y/o en general sobre los parámetros de codificación tal como si se debe usar inter- o intra-predicción para un bloque y con qué ajustes.

La unidad de intra-predicción 109 recibe como entrada un bloque de un cuadro o imagen actual para que se intra-codifiquen y una o varias muestras de referencia de un área ya reconstruida del cuadro actual. La intra-predicción luego describe los píxeles de un bloque actual del cuadro actual en términos de una función de muestras de referencia del cuadro actual. La unidad de intra-predicción 109 emite un bloque de predicción para el bloque actual, en donde este bloque de predicción minimiza ventajosamente la diferencia entre el bloque actual a codificar y su bloque de predicción, es decir, minimiza el bloque residual. La minimización del bloque residual puede basarse, por ejemplo en un procedimiento de optimización de la velocidad de distorsión. En particular, el bloque de predicción se obtiene como una interpolación direccional de las muestras de referencia. La dirección puede determinarse mediante la optimización de la distorsión de velocidad y/o calculando una medida de similitud como se mencionó anteriormente en relación con la predicción cruzada.

La diferencia entre el bloque actual y su predicción, es decir, el bloque residual, se transforma por la unidad de transformación 101. Los coeficientes de transformación se cuantifican por la unidad de cuantificación 102 y se codifican por entropía por la unidad de codificación entrópica 103. El flujo de bits de video codificado generado de este modo comprende bloques intra-codificados y bloques inter-codificados y la señalización correspondiente (tal como la indicación de modo, la indicación de vector de movimiento y/o la dirección de inter-predicción). La unidad de transformación 101 puede aplicar una transformada lineal tal como una transformada de coseno discreta o de Fourier (DFT). Esta transformada en el dominio de frecuencia espacial proporciona la ventaja de que los coeficientes resultantes tienen valores típicamente más altos en las frecuencias más bajas. Por lo tanto, después de un escaneo de coeficientes efectivo (tal como zig-zag) y cuantificación, la secuencia de valores resultante tiene típicamente algunos valores más grandes al principio y

termina con una serie de ceros. Esto permite una codificación más eficiente. La unidad de cuantificación 102 realiza la compresión con pérdida real reduciendo la resolución de los valores de coeficiente. La unidad de codificación entrópica 103 luego asigna valores de coeficientes a las palabras de código binarias para producir un flujo de bits. El codificador entrópico también codifica la información de señalización (no se muestra en la figura 1).

La figura 2 muestra un decodificador de video 200. El decodificador de video 200 comprende particularmente una memoria intermedia de imagen de referencia 207 y una unidad de intra-predicción 209, que es una unidad de predicción de bloque. La memoria intermedia de imagen de referencia 207 se adapta para almacenar al menos un cuadro de referencia reconstruido a partir del flujo de bits de video codificado, este cuadro de referencia que es diferente de un cuadro actual (cuadro actualmente decodificado) del flujo de bits de video codificado. La unidad de intra-predicción 209 se configura para generar un bloque de predicción, que es una estimación del bloque que se va a decodificar. La unidad de intra- predicción 209 se configura para generar esta predicción con base en muestras de referencia que se obtienen de la memoria intermedia de imágenes de referencia 207.

El decodificador 200 se adapta para decodificar el flujo de bits de video codificado generado por el codificador de video 100, y preferiblemente tanto el decodificador 200 como el codificador 100 generan predicciones idénticas para que el bloque respectivo se codifique/decodifique. Las características de la memoria intermedia de imagen de referencia 207 y la unidad de intra-predicción 209 son similares a las características de la memoria intermedia de imágenes de referencia 107 y la unidad de intra-predicción 109 de la figura 1.

El decodificador de video 200 comprende unidades adicionales que también están presentes en el codificador de video 100 tal como, por ejemplo una unidad de cuantificación inversa 204, una unidad de transformación inversa 205 y una unidad de filtración de bucle 206, que corresponden respectivamente a la unidad de cuantificación inversa 104, la unidad de transformación inversa 105 y la unidad de filtración de bucle 106 del codificador de video 100.

Una unidad de decodificación entrópica 203 se adapta para decodificar el flujo de bits de video codificado recibido y para obtener correspondientemente coeficientes de transformación residual cuantificados e información de señalización. Los coeficientes de transformación residual cuantificados se alimentan a la unidad de cuantificación inversa 204 y a una unidad de transformación inversa 205 para generar un bloque residual. El bloque residual se agrega a un bloque de predicción y la adición se alimenta a la unidad de filtración de bucle 206 para obtener el video decodificado. Los cuadros del video decodificado pueden almacenarse en la memoria intermedia de imágenes de referencia 207 y servir como un cuadro de referencia para inter-predicción.

Generalmente, las unidades de intra-predicción 109 y 209 de las figuras 1 y 2 pueden usar muestras de referencia de un área ya codificada para generar señales de predicción para bloques que deben codificarse o decodificarse.

La unidad de decodificación entrópica 203 recibe como entrada el flujo de bits codificado. En general, el flujo de bits se analiza primero, es decir, los parámetros de señalización y los residuos se extraen del flujo de bits. Típicamente, la sintaxis y la semántica del flujo de bits se definen mediante una norma para que los codificadores y decodificadores puedan funcionar de manera interoperable. Tal como se describe en la sección de antecedentes anterior, el flujo de bits codificado no solo incluye los residuos de predicción. En caso de predicción de movimiento compensado, una indicación de vector de movimiento también se codifica en el flujo de bits y se analiza de la misma en el decodificador. La indicación de vector de movimiento se puede dar por medio de una imagen de referencia en la que se proporciona el vector de movimiento y por medio de las coordenadas del vector de movimiento. Para codificar de manera eficiente la imagen de referencia, el códec H.265 (ITU-T, H265, Series H: Audiovisual and multimedia systems: High Efficient Video Coding) proporciona una lista de imágenes de referencia asignadas para enumerar los índices de los marcos de referencia respectivos. El cuadro de referencia se señala en el flujo de bits al incluir en el mismo el índice de lista asignado correspondiente. Esta lista puede definirse en la norma o indicarse al comienzo del video o un conjunto de varios cuadros. Se observa que en H.265 hay dos listas de imágenes de referencia definidas, llamadas L0 y L1. La imagen de referencia se señala en el flujo de bits indicando la lista (L0 o L1) e indica un índice en esa lista asociado con la imagen de referencia deseada. Proporcionar dos o más listas puede tener ventajas para una mejor compresión. Por ejemplo, L0 se puede usar tanto para segmentos inter-predichos unidireccionalmente como para segmentos inter-predichos bidireccionalmente, mientras que L1 solo se puede usar para segmentos inter-predichos bidireccionalmente. Sin embargo, en general, la presente divulgación no se limita a algún contenido de las listas L0 y L1.

El vector de movimiento puede señalizarse directamente mediante las coordenadas en la imagen de referencia. Alternativamente, como también se especifica en H.265, se puede construir una lista de vectores de movimiento candidatos y se puede transmitir un índice asociado en la lista con el vector de movimiento particular.

Los vectores de movimiento del bloque actual generalmente se correlacionan con los vectores de movimiento

de bloques vecinos en la imagen actual o en las imágenes codificadas anteriores. Esto se debe a que es probable que los bloques vecinos correspondan al mismo objeto en movimiento con un movimiento similar y no es probable que el movimiento del objeto cambie abruptamente con el tiempo. En consecuencia, el uso de los vectores de movimiento en bloques vecinos como predictores reduce el tamaño de la diferencia del vector de movimiento señalado. Los MVP generalmente se derivan de vectores de movimiento ya decodificados de bloques vecinos espaciales o de bloques vecinos temporales en la imagen co-ubicada. En H.264/AVC, esto se realiza haciendo una mediana inteligente de componentes de tres vectores de movimiento espacialmente vecinos. Con este enfoque, no se requiere señalización del predictor. Los MVP temporales de una imagen co-ubicada solo se consideran en el llamado modo directo temporal de H.264/AVC. Los modos directos H.264/AVC también se utilizan para derivar otros datos de movimiento que los vectores de movimiento. Por lo tanto, se relacionan más con el concepto de fusión de bloques en HEVC. En HEVC, el enfoque de derivar implícitamente el MVP se reemplazó por una técnica conocida como competencia de vectores de movimiento, que señala explícitamente qué MVP de una lista de MVP se usa para la derivación de vectores de movimiento. La estructura de bloques de árbol cuádruple de codificación variable en HEVC puede dar como resultado que un bloque tenga varios bloques vecinos con vectores de movimiento como candidatos potenciales a MVP. Tomando como ejemplo al vecino izquierdo, en el peor de los casos, un bloque de predicción de 64x64 luma podría tener 16 bloques de predicción de 84 luma a la izquierda cuando un bloque de árbol de codificación de 64x64 luma no se divide más y el izquierdo se divide a la profundidad máxima. Se introdujo la predicción avanzada de vectores de movimiento (AMVP) para modificar la competencia de vectores de movimiento para tener en cuenta esta estructura de bloque flexible. Durante el desarrollo de HEVC, el diseño inicial de AMVP se simplificó significativamente para proporcionar un buen equilibrio entre la eficiencia de codificación y un diseño amigable para la implementación. El diseño inicial de AMVP incluyó cinco MVP de tres clases diferentes de predictores: tres vectores de movimiento de vecinos espaciales, la mediana de los tres predictores espaciales y un vector de movimiento modificado en escala de un bloque co-ubicado, temporalmente vecino. Además, la lista de predictores se modificó reordenando para colocar el predictor de movimiento más probable en la primera posición y eliminando candidatos redundantes para asegurar una sobrecarga mínima de señalización. El diseño final de la construcción de lista de candidatos AMVP incluye los siguientes dos candidatos MVP: a) hasta dos MVP candidatos espaciales que se derivan de cinco bloques vecinos espaciales; b) un MVP candidato temporal derivado de dos bloques temporales ubicados conjuntamente cuando ambos MVP candidatos espaciales no están disponibles o son idénticos; y c) vectores de movimiento cero cuando los candidatos espaciales, temporales o ambos no están disponibles. Los detalles sobre la determinación del vector de movimiento se pueden encontrar en el libro de V. Sze et al (Ed.), High Efficiency Video Coding (HEVC): Algorithms and Architectures, Springer, 2014, en particular en el capítulo 5.

Los bloques utilizados para generar la lista de candidatos se muestran en la figura 4: el bloque C1 (cero) co-ubicado, el vecino temporal C1 y los vecinos espaciales A0, A1, B0, B1, B2, adyacentes al bloque actual.

Para mejorar aún más la estimación del vector de movimiento sin un aumento adicional en la sobrecarga de señalización, puede ser beneficioso refinar aún más el vector de movimiento derivado en el lado de codificador y proporcionado en el flujo de bits. El refinamiento de vector de movimiento se puede realizar en el decodificador sin ayuda del codificador. El codificador en su bucle de decodificador puede emplear el mismo refinamiento para obtener imágenes de referencia correspondientes. El refinamiento se realiza determinando una plantilla, determinando un espacio de búsqueda y encontrando en el espacio de búsqueda una porción de imagen de referencia que mejor se adapta a la plantilla. La mejor posición de la porción coincidente determina el mejor vector de movimiento que luego se utiliza para obtener el predictor del bloque actual, es decir, el bloque actual que se reconstruye.

Cuando se aplica el refinamiento de vector de movimiento en el lado de decodificador, también el bucle de decodificador en el lado de codificador tiene que implementar el refinamiento de vector de movimiento para generar las mismas imágenes de referencia que se generarán y usarán en el lado de decodificador. La etapa de derivación de vector de movimiento que precede al refinamiento de vector de movimiento generalmente requiere el conocimiento de los vectores de movimiento de los bloques que rodean espacialmente el bloque actual que se decodifica, en particular en los casos en que la lista de vectores de movimiento candidatos se usa para seleccionar la estimación del vector de movimiento que se va a refinar de forma adicional. La lista de vectores de movimiento candidatos para el bloque actual se crea con base en los vectores de movimiento de los bloques ya decodificados y adyacentes espacial o temporalmente al bloque actual. Esta dependencia de la determinación de vector de movimiento en la disponibilidad de los vectores de movimiento de los bloques vecinos dificulta aún más la paralelización de la codificación y/o decodificación. Además, esta dependencia también reduce la resiliencia de errores de la secuencia codificada. En particular, para realizar el refinamiento de vector de movimiento, es necesario tener imágenes de referencia decodificadas correctamente con el fin de hacer coincidir las plantillas. Si la imagen de referencia incluye errores, se propagan aún más. Se observa que la coincidencia de plantillas se usa típicamente para el refinamiento de vector de movimiento. Sin embargo, la presente divulgación también es aplicable a casos en los que la coincidencia de plantillas se aplica a la determinación del vector de movimiento.

La figura 3 muestra un ejemplo de una estructura de las unidades de inter-predicción mostradas en las figuras

1 y 2. Al principio, los datos de movimiento se determinan 310. Esto se puede realizar como se describe anteriormente por AMVP 315 seleccionando vectores de movimiento de una lista de vectores de movimiento candidatos. Tan pronto como se determinan los vectores de movimiento, puede tener lugar la interpolación fraccionaria 320 (especialmente si el vector o vectores de movimiento apuntan a una posición de píxel fraccionaria) y se determinan los predictores para los respectivos vectores de movimiento (330: dirección 0 y 335: dirección 1). Con base en la determinación de los predictores en las direcciones respectivas (es decir, imágenes de referencia), se forma una plantilla para coincidencia realizada durante el bloque de refinamiento de vector de movimiento de decodificador 340. Los vectores de movimiento refinados entonces se ponderan 350 para obtener 360 una predicción de movimiento compensado del bloque actual.

La figura 5 ilustra el procesamiento en el decodificador, cuando se aplica el refinamiento de vector de movimiento. En particular, las unidades de procesamiento PU1, PU2, PU3, etc. se analizan por un motor de análisis uno tras otro como se muestra en la parte superior de la figura 5. El motor de análisis puede corresponder al decodificador entrópico 203 de la figura 2. Cada unidad de procesamiento analizada que se va a procesar por la interpredicción para reconstruir la textura del bloque de imagen se proporciona al bloque 210 de inter-predicción, que se ilustra mediante las flechas verticales. Al observar la unidad de procesamiento PU2 (columna derecha de flechas), se puede ver que debido a la aplicación del refinamiento, se vuelve casi imposible procesamiento paralelo de PU1 y PU2. Para construir una lista de fusión (en general, una lista de vectores de movimiento candidatos basados en los bloques ya decodificados), deben conocerse los vectores de movimiento de los bloques vecinos tal como PU1. Dado que el refinamiento de vector de movimiento puede basarse en la coincidencia de plantillas, que es computacionalmente compleja, y los vectores de movimiento se conocen solo después del refinamiento, PU2 no puede iniciar la derivación de vector de movimiento hasta que se hayan procesado todos los bloques cuyos vectores de movimiento se utilizan en la lista de fusión (PU1 en la figura 5). En particular, para PU1 la construcción de lista de fusión, la selección de vector de movimiento inicial, por ejemplo, mediante el uso del índice de la lista de fusión extraído del flujo de bits, así como el refinamiento se deben realizar antes de que el o los vectores de movimiento refinados se pasen a procesamiento de PU2 para generar la lista de fusión. De acuerdo con la figura 5, la reconstrucción de la textura ya está completa durante la coincidencia de plantillas realizada aquí en el paso de refinamiento. En consecuencia, después del refinamiento, el vector de movimiento PU1 puede pasarse al procesamiento de PU2. La filtración no cambia el vector de movimiento y simplemente se aplica a la PU1 reconstruida para mejorar aún más su calidad. La filtración puede ser, por ejemplo, una filtración de desbloqueo o cualquier filtración adaptativa.

Como se puede ver en este ejemplo, el procesamiento de diferentes unidades de procesamiento es casi secuencial. Además, el refinamiento de vector de movimiento requiere para su funcionamiento un espacio de búsqueda formado por muestras o muestras fraccionarias de la imagen de referencia. Por lo tanto, si falta la imagen de referencia o está decodificada con distorsiones, esto puede influir en términos de calidad en la decodificación del bloque actual. Por lo tanto, la decodificación que se muestra en la figura 5 que ilustra el procesamiento de decodificador inter-predicción que hace uso de o el refinamiento de vector de movimiento no permite la paralelización eficiente para diferentes PU e introduce algunas degradaciones adicionales causadas por la imagen de referencia faltante o decodificada incorrectamente.

En otras palabras, los procesos de derivación y refinamiento de vector de movimiento de lado de decodificador pueden producir 1) problemas de resiliencia de errores y problemas de paralelización. Ciertos casos de uso requieren que la propagación de errores en la dirección espacial y temporal deba ser controlable (puede limitarse dentro de una región conocida o puede recuperarse tomando ciertas medidas). Los métodos de la técnica anterior tienen el problema de que, si hay un error en la transmisión, el error de reconstrucción se propaga sin control en las direcciones temporales y espaciales.

Con respecto al problema del procesamiento paralelo de lado de decodificador, las herramientas de derivación y refinamiento de MV de lado de decodificador son computacionalmente intensivas ya que requieren operaciones de coincidencia de plantillas dentro de un gran espacio de puntos candidatos. Por lo tanto, es necesario poder realizar las operaciones de derivación y refinamiento en paralelo para que las PU vecinas aumenten el rendimiento.

El término PU (unidad de procesamiento) aquí se refiere a una unidad de imagen como el bloque que se procesa mediante la inter-predicción. Las PU de una imagen también pueden tener diferentes tamaños.

Los códecs actuales como H.265/HEVC proporcionan algunas características de resiliencia, como desactivación de la predicción de vector de movimiento para ciertos casos de uso. Por ejemplo, el vector de movimiento de un bloque colocado temporalmente no se usa para predicción de MV, si un indicador en el flujo de bits lo indica. Además, se activa la estimación de fusión paralela para la decisión de codificador. La sección 5.2.2.5 "Merge Estimation Regions for Parallel Merge Mode Estimation" of the book "High Efficiency Video Coding (HEVC), Vivienne Sze, Madhukar Budagavi, Gary J. Sullivan", Springer, 2014 explica la técnica en detalle. En particular, se define una ventana (región de estimación de movimiento, MER), donde los bloques que caen en la región no pueden heredar información de movimiento entre sí. En otras palabras, si dos bloques

(bloques actuales y vecinos) están en el mismo MER, la información de movimiento del bloque vecino no se incluye en la lista de fusión del bloque actual, para mejorar la capacidad de procesamiento en paralelo.

5 Con el fin de mejorar la posibilidad de paralelización y, al mismo tiempo, proporcionar una mayor resiliencia de errores, de acuerdo con la presente descripción, solo se aplica coincidencia de plantillas para determinar los vectores de movimiento de un bloque actual, si el bloque vecino no ha obtenido el vector de movimiento por coincidencia de plantillas.

10 En consecuencia, dado que los bloques que usan la coincidencia de plantillas, no se usan más para decodificar bloques adyacentes, los bloques adyacentes pueden decodificarse durante el procesamiento de los bloques decodificados utilizando la coincidencia de plantillas. Además, dado que los bloques adyacentes no usan bloques reconstruidos por coincidencia de plantillas, pueden decodificarse incluso si falta o está dañada la imagen de referencia necesaria para la coincidencia de plantillas.

15 La invención también puede limitar la predicción a partir de bloques vecinos espaciales, si el bloque vecino aplica derivación o refinamiento de movimiento con base en la coincidencia de plantillas. Puede limitar la predicción de un bloque vecino temporal, solo si el bloque aplica derivación o refinamiento y estas dos técnicas se pueden aplicar juntas. El HEVC actualmente proporciona dos enfoques diferentes para ayudar a la resiliencia de errores y la paralelización como se mencionó anteriormente. Desactiva completamente la
20 predicción o herencia de información de movimiento dentro de una ventana predefinida. Como resultado, intercambia ganancia de codificación con capacidad de procesamiento paralelo.

25 En algunas realizaciones, la información de movimiento de un bloque vecino se reemplaza con una versión no óptima para que se utilice por el bloque actual para predicción (posiblemente si se encuentran dentro de la misma ventana). Ya que la predicción no está completamente desactivada, la pérdida de codificación es mucho menor. La presente invención se aplica a bloques vecinos que aplican derivación o refinamiento de movimiento. Si un bloque vecino no utiliza el proceso de derivación o refinamiento, la información de movimiento se puede heredar (o utilizada como predictor) por el bloque actual.

30 Un aparato de ejemplo de acuerdo con una realización para determinar una predicción de un bloque actual, para la reconstrucción del bloque actual, comprende: una unidad de predicción para detectar si o no un vector de movimiento de un bloque adyacente al bloque actual se ha determinado por coincidencia de plantillas. Si se detecta negativamente, derivar predicción del bloque actual de acuerdo con el bloque adyacente. Por otro lado, si se detecta positivamente, no derivar predicción del bloque actual de acuerdo con el bloque adyacente.

35 Esto se ilustra en la figura 6, en la cual la unidad de inter-predicción tal como la 210 de la figura 2 implementa además la detección de si usar un bloque vecino para predicción. Esto puede lograrse mediante la unidad de decisión de bloque 660 añadida al dispositivo de predicción 600 correspondiente al dispositivo de predicción 300 de la figura 3.

40 Ejemplo 1: Vector de reemplazo

De acuerdo con un primer ejemplo, la invención puede aplicarse a la determinación de vectores de movimiento con base en una lista de candidatos formados a partir de los vectores de movimiento de bloques vecinos. Por
45 lo tanto, el aparato mencionado anteriormente para determinar la predicción de un bloque actual se configura además para determinar un vector de movimiento para determinar la predicción cruzada de un bloque actual. Para la reconstrucción del bloque actual, la unidad de predicción comprende una unidad de determinación de lista de vectores de movimiento 615A, 615B para formar una lista de vectores de movimiento candidatos para el bloque actual, la formación de la lista incluye:

- 50 - detectar, mediante la unidad de decisión de bloque 660, si el vector de movimiento de un bloque adyacente se ha determinado por la coincidencia de plantillas,
- si se detecta negativamente, incluir el vector de movimiento del bloque adyacente como candidato en la lista,
- 55 - si se detecta positivamente, no incluir el vector de movimiento del bloque adyacente como candidato en la lista.

60 El aparato puede comprender además una unidad de determinación de vector de movimiento 620A, 620B para seleccionar una estimación de un vector de movimiento de la lista y determinar el vector de movimiento del bloque actual de acuerdo con la estimación.

En particular, la unidad de determinación de lista 615A, 615B puede configurarse para, si se detecta positivamente, incluir un vector de movimiento de reemplazo como candidato en la lista.

65 Como se puede ver en la figura 6, la derivación de vector de movimiento incluye el bloque de fusión del bloque

de predicción entre imágenes, así como predicción avanzada de vector de movimiento. Sin embargo, la presente divulgación también es aplicable si solo se proporciona uno de estos enfoques/bloques, o si se utiliza cualquier otro enfoque de derivación de vector de movimiento, que se basa en dos pasos, es decir, generación de lista de vectores de movimiento y selección de un vector de movimiento de la lista. La unidad de decisión de bloque 660 recibe en su entrada la información 605 de tipo/modo de predicción del bloque vecino (previamente decodificado y adyacente espacial o temporalmente de forma inmediata o diagonal). Con base en esto, determina si el vector de movimiento del bloque actual se usará o no para construir la lista de MV (615A y/o 615B).

La unidad AMVP recibe en su entrada 602 información sobre la diferencia entre la predicción de vector de movimiento y el vector de movimiento de acuerdo con lo determinado por el codificador y señalado en el flujo de bits. La lista de MV 615B se genera de acuerdo con la decisión de la unidad de decisión de bloque 660 y la selección de candidatos 620B se realiza seleccionando la predicción de vector de movimiento de la lista. De manera similar, la fusión de bloques de predicción entre imágenes (IPBM) incluye generación de la lista de fusión 615A de acuerdo con el control por la unidad de decisión de bloque 660 y selección 615A del mejor candidato de la lista de fusión. La interpolación 320 puede, pero en general no tiene que realizarse. Entonces, la predicción se calcula con base en un solo vector de movimiento (en este caso, solo se usa una de las dos unidades 330 y 335) o con base en 2 MV en las unidades 330 y 335 respectivamente. Se observa que, en general, solo se puede implementar una de las unidades 330 y 335, si no se soporta bi-predicción. Estos bloques 320-360 funcionan como en los descritos con referencia a la figura 3.

La figura 7 ilustra un ejemplo, en el que también se incluye en el aparato una unidad 340 de refinamiento de vector de movimiento para aplicar un refinamiento de la estimación seleccionada del vector de movimiento. Además, la unidad de determinación de lista de vectores de movimiento candidatos 615 se configura para, si se detecta positivamente, incluir en la lista una estimación del vector de movimiento del bloque adyacente obtenido por la coincidencia de plantillas antes de aplicar el refinamiento y la unidad de determinación de vector de movimiento 310 se configura además para determinar el vector de movimiento del bloque adyacente de acuerdo con la estimación del vector de movimiento del bloque adyacente después del refinamiento. En otras palabras, en este ejemplo, la versión no refinada del vector de movimiento, en términos generales es la estimación del vector de movimiento antes de la coincidencia de plantillas, se almacena en una memoria para su posterior uso por el proceso de reconstrucción de una siguiente unidad de procesamiento. Las versiones refinadas de los vectores de movimiento se utilizan para el proceso de reconstrucción de muestras del bloque de codificación actual, posiblemente seguido de un proceso de filtración.

La figura 7 muestra análisis de las unidades de procesamiento de manera similar al ejemplo de la figura 4. Sin embargo, cuando se procesa PU1, el vector de movimiento sin refinar que se selecciona para PU1 adyacente a PU2 se envía al proceso de procesamiento de PU2. En particular, PU2 utiliza el vector de movimiento no refinado de PU1 para formar su lista de vectores de movimiento candidatos. Al mismo tiempo, el procesamiento de PU1 continúa mediante refinamiento de vector de movimiento, reconstrucción de textura (predicción del bloque actual con base en el vector de movimiento refinado) y filtración. Se observa que la etapa de filtración puede omitirse. Como se puede ver, la lista de vectores de movimiento candidatos de PU2 ahora se puede formar y se puede realizar selección y refinamiento de vector de movimiento mientras PU1 aún se procesa. En consecuencia, es posible paralelización en el nivel de procesamiento entre predicciones para las unidades de procesamiento vecinas. Al mismo tiempo, dado que el vector de movimiento seleccionado para PU1 antes del refinamiento no depende de la disponibilidad/corrección de la imagen de referencia. Por lo tanto, no tiene lugar propagación de errores de PU1 a PU2, lo que lleva a una mayor resiliencia de errores.

En otras palabras, en la figura 7, la PU1 utiliza la salida del proceso de derivación/refinamiento para procesos de reconstrucción de textura y filtración. La PU2 (que es vecina de PU1) no utiliza la salida de la derivación/refinamiento (de PU1) para predicción de MV. En su lugar, utiliza una versión no óptima (que se puede obtener antes de que se complete el proceso de refinamiento/derivación) para predicción de MV.

Se observa que una de las ventajas proporcionadas por el ejemplo 1 es que se activa paralelización. Sin embargo, no cada codificador/decodificador necesariamente debe hacer uso de la paralelización. Otra ventaja es que el codificador/decodificador puede seleccionar si se realizará o no paralelización. Cuando se realiza la paralelización, la unidad de determinación de lista de vectores de movimiento se configura para formar la lista para el bloque actual, mientras que la unidad de refinamiento de vector de movimiento aplica el refinamiento al bloque adyacente.

La parte anterior del primer ejemplo no distingue entre el tipo de vecindad, en particular si los bloques vecinos son inmediatamente adyacentes al bloque actual dentro del dominio temporal o espacial. Puede proporcionar algunas ventajas adicionales para hacer esta distinción. En particular, los vectores de movimiento de los vecinos temporalmente anteriores (en el orden de decodificación) están típicamente disponibles cuando se forma la lista de vectores de movimiento para el bloque actual. En consecuencia, si solo se enfoca en la ventaja de paralelización, puede ser beneficioso incluir los bloques temporalmente adyacentes en la lista de vectores de movimiento, incluso si se obtuvieron aplicando coincidencia de plantillas y, más específicamente, en

términos de este ejemplo, si se refinaron.

Por lo tanto, correspondientemente, la unidad de determinación de lista de vectores de movimiento puede configurarse para, si se detecta positivamente,

5

- si el bloque adyacente se encuentra en la misma imagen que el bloque actual, incluir en la lista una estimación del vector de movimiento del bloque adyacente obtenido por la coincidencia de plantillas antes de aplicar el refinamiento, y

10

- si el bloque adyacente se encuentra en una imagen distinta del bloque actual, incluir en la lista el vector de movimiento del bloque adyacente después de aplicar el refinamiento al bloque adyacente.

Dado que la imagen de referencia temporal se reconstruye antes de la imagen actual, hacer referencia a los vectores de movimiento refinados en la imagen de referencia temporal no ralentizaría la reconstrucción de la imagen actual. Cada unidad de procesamiento que aplica el refinamiento de vector de movimiento almacena dos candidatos para una predicción posterior, es decir: vector de movimiento no refinado que se va a usar por vecinos espaciales para predicción de vector de movimiento por unidades de procesamiento posteriores (en orden de decodificación) y un vector de movimiento refinado que se va a utilizar si predicción de movimiento temporal por los bloques posteriores, así como por compensación de movimiento y posiblemente filtración de la unidad de procesamiento actual. Como se mencionó anteriormente, el uso del reemplazo en lugar de realizar la coincidencia de plantillas puede disminuir ligeramente la eficiencia de codificación.

15

20

La figura 8 ilustra un método de ejemplo que no forma parte de la invención que ilustra el enfoque descrito anteriormente. En particular, en el paso S810, se determina si el bloque actual y uno de sus vecinos se encuentra dentro de la misma ventana. La ventana puede ser, por ejemplo, una ventana espacial. En este ejemplo, la ventana puede ser la imagen completa. En consecuencia, se puede probar si los bloques actuales y vecinos están dentro de la misma ventana. Si no ("no" en el paso S810), el bloque actual usa el vector de movimiento refinado para la predicción de vector de movimiento, por ejemplo, el vector de movimiento refinado del bloque vecino se incluye en la lista de vectores de movimiento candidatos en el paso S850.

25

30

De lo contrario, si el bloque vecino y el bloque actual se encuentran dentro de la ventana, en el paso S820 se prueba si el bloque vecino aplica o no refinamiento o derivación de vector de movimiento utilizando la coincidencia de plantillas. Si "no", es decir, si el bloque vecino no tiene un vector de movimiento determinado por la coincidencia de plantillas, entonces sigue el paso S50, lo que significa que se realiza el refinamiento para el bloque actual. Por otro lado, si el bloque vecino aplica el refinamiento o la derivación de vector de movimiento con coincidencia de plantillas, en el paso S830 se prueba adicionalmente si el bloque actual y uno de sus vecinos pertenecen a la misma imagen. En consecuencia, si el bloque actual y el bloque vecino no están dentro de la misma imagen ("no" en el paso S830), el bloque actual usa el vector de movimiento refinado para predicción de vector de movimiento, por ejemplo, el vector de movimiento refinado del bloque temporalmente adyacente se incluye en la lista de vectores de movimiento candidatos en el paso S850. Si, por otro lado, en el paso S830 la respuesta es "sí", en el paso S840, el bloque actual usa un vector de movimiento de reemplazo para predicción de vector de movimiento.

35

40

En el ejemplo anterior, los pasos S810 y S830 parecen probar la misma condición. Sin embargo, en el siguiente ejemplo, se analizan otras posibilidades de ventana espacial para el paso 810. Además, se observa que la prueba en el paso S830 también puede probar una ventana temporal, es decir, probar si el bloque vecino en el orden de decodificación está ubicado a menos de k imágenes en orden de decodificación de la imagen actual en la que se encuentra el bloque actual. Se observa que decodificación se considera en general lo mismo que un orden de codificación. En otras palabras, el orden de decodificación está dado por el orden de codificación definido por el lado de codificador y la estructura de flujo de bits generada. Por lo tanto, al referirse al orden de decodificación en este ejemplo, se podría hacer referencia al orden de codificación.

45

50

En el ejemplo anterior de la figura 7, el vector de movimiento de reemplazo se ha determinado por el vector de movimiento no refinado. Sin embargo, la presente descripción no se limita a este enfoque. Más bien, el reemplazo puede definirse de cualquier manera.

55

Sin embargo, el vector de movimiento de reemplazo no es necesariamente un vector de movimiento del bloque vecino antes del refinamiento. Por ejemplo, en una realización de ejemplo, el vector de movimiento de reemplazo es un vector de movimiento de un bloque vecino al bloque actual y diferente del bloque adyacente.

60

La figura 10 ilustra una PU actual que tiene una PU vecina denominada A1 y A1 tiene una PU vecina denominada X1. Si el bloque vecino A1 aplica derivación/refinamiento de vector de movimiento (coincidencia de plantillas), entonces se construye una lista de vectores de movimiento candidatos para PU A1 usando los vecinos espaciales y/o temporales de A1 (por ejemplo, X1). Un vector de movimiento candidato para la PU actual se selecciona o calcula a partir de la lista de candidatos construida para el bloque vecino A1. El candidato seleccionado o calculado A1_MV_replacement puede diferir del vector de movimiento que saldrá de la

65

determinación del vector de movimiento para A1. Aquí, selección significa selección de una lista de posibles candidatos. El cálculo de los vectores de movimiento obtiene el vector de movimiento de otra manera, por ejemplo, promediando los vectores de movimiento de más de un candidato o calculando el vector de movimiento de cualquier otra manera con base en los MV candidatos o sin referirse a los candidatos.

5 Ventajosamente, A1_MV_replacement no se ha obtenido por coincidencia de plantillas. Sin embargo, se observa que dado que X1 no es un vecino directo a la PU actual, de acuerdo con una implementación, el reemplazo de A1_MV también se puede determinar utilizando la coincidencia de plantillas, ya que es probable que A1_MV_replacement esté disponible al decodificar la PU actual, de modo que todavía es posible procesamiento paralelo de la corriente PU y, por ejemplo, A1.

10 De acuerdo con un ejemplo, X1 es un bloque adyacente de A1 en su lado izquierdo, pero X1 no es adyacente a la unidad de procesamiento actual. Sin embargo, se puede usar cualquier otro bloque en la proximidad de A1 o en la vecindad temporal de A1. El vector de movimiento de X1 se puede usar como A1_MV_replacement.

15 En general, una lista de candidatos (conjunto) de la PU actual se forma enumerando en ella los vectores de movimiento de algunas PU vecinas. Es una PU vecina espacial y/o temporalmente de la cual el vector de movimiento se debe incluir en la lista de candidatos de la PU actual se determina mediante la coincidencia de plantillas, luego se incluye un vector de reemplazo en el conjunto candidato de la PU actual en su lugar. Este vector de reemplazo, de acuerdo con un ejemplo, es un vector de movimiento de la lista de candidatos de esa

20 PU espacial y/o temporalmente vecina. El vector de reemplazo puede ser en general un vector de movimiento ya refinado o un vector de movimiento determinado sin coincidencia de plantillas (por ejemplo, por predicción de vector de movimiento basada en señalización de flujo de bits).

25 La figura 9 muestra otra realización de ejemplo del vector de movimiento de reemplazo. De acuerdo con esta realización de ejemplo, cuando se construye la lista de candidatos para la PU actual, un vector de movimiento de un bloque vecino que se va a incluir en la lista de candidatos pero se determina usando la coincidencia de plantillas se reemplaza por un vector de movimiento del siguiente vector de movimiento candidato que se va a incluir en la lista. Por ejemplo, la figura 9 muestra la PU actual con vecinos A0 a Am + 1 en el lado izquierdo y vecinos B-1 a Bn + 1 en la parte superior.

30 Como se puede ver en la figura 9, en general (en cualquiera de los ejemplos divulgados en el presente documento), las PU que se van a procesar pueden tener diferentes tamaños. En consecuencia, la PU actual puede tener varios vecinos espaciales y también varios vecinos temporales, e incluso al menos parcialmente ubicados (en imágenes previamente decodificadas).

35 Además, de acuerdo con una implementación que también puede ser aplicable a cualquiera de los ejemplos y realizaciones divulgados en el presente documento, la construcción de lista de candidatos para cada PU se realiza de una manera predefinida, por ejemplo, definida en una norma. Por ejemplo, la lista de candidatos se puede construir mediante la inclusión de vectores de movimiento de los vecinos actuales de la PU de acuerdo

40 con un orden predefinido. La lista de candidatos puede tener un tamaño fijo. Esta regla para crear la lista de candidatos permite que el codificador y el decodificador funcionen de manera compatible. Por ejemplo, el orden predefinido puede ser los vectores de movimiento A0 a Am + 1 seguidos por los vectores de movimiento de B-1 a Bn + 1 seguidos por los vectores de movimiento C0 a Cp en una imagen previamente decodificada (directamente adyacente). Aquí, m es el número de vecinos del lado izquierdo, n es el número de vecinos superiores y p es el número de vecinos temporales. Sin embargo, se observa que la manera en que se construye la lista de candidatos no está necesariamente completamente definida en la norma. Puede estar determinado al menos parcialmente por la señalización relacionada con la secuencia de video dentro del flujo de bits. Además, de acuerdo con la regla predefinida, el vector de movimiento de todos los vecinos mostrados

45 en la figura 9 y/o los vecinos temporales no se incluyen necesariamente en la lista. La regla puede definir solo varios de los vecinos y puede, pero no necesariamente, incluir vecinos temporales. Además, se observa que la lista no tiene que incluir directamente los vectores de movimiento de los bloques vecinos. También puede incluir vectores de movimiento calculados a partir de los vectores de movimiento de los bloques vecinos (por ejemplo, un promedio de dos vectores de movimiento de vecinos mutuamente adyacentes o similares).

50 En esta realización de ejemplo, la lista de candidatos se construye para la PU actual. Si un vector de movimiento de Am + 1 debe incluirse en la lista (de acuerdo con la regla predefinida) pero se ha determinado mediante la coincidencia de plantillas, entonces, en lugar del vector de movimiento de Am + 1, el vector de movimiento de la PU después de Am + 1 en la lista de candidatos de la PU actual se utiliza como un reemplazo para el vector de movimiento de Am + 1. En este ejemplo (ver la secuencia de bloques mencionada anteriormente), el vector

55 de reemplazo sería B-1. Como se mencionó anteriormente, la regla predefinida puede definir un orden diferente, como Am + 1 a A0, B-1 a Bn + 1, C0. En este caso, el siguiente bloque después de Am + 1 sería Am, de modo que el vector de movimiento de Am + 1 se reemplazaría por el vector de movimiento de Am en la lista de candidatos para la PU actual.

60 Se observa que en caso de que todos los bloques vecinos usen la coincidencia de plantillas para derivar sus respectivos vectores de movimiento y, por lo tanto, el vector de movimiento del bloque actual no se puede

determinar tomando el vector de movimiento de otro vecino, entonces, de acuerdo con una realización, el bloque actual no debe aplicar coincidencia de plantillas para la derivación o refinamiento de vector de movimiento. Alternativamente, en este caso, se puede usar un conjunto predefinido de candidatos MV para construir la lista de predictores MV candidatos. Este conjunto predefinido podría incluir MV que son cero en los ejes x e y. De manera similar, si no hay ningún vecino del bloque vecino que pueda usarse para determinar el vector de movimiento, el bloque actual se predecirá de manera diferente. Como un ejemplo, podría predecirse usando intra-predicción en lugar de inter-predicción.

En las implementaciones de ejemplo anteriores, un vector de movimiento candidato de una unidad de procesamiento actual determinada por coincidencia de plantillas se reemplaza por un vector de reemplazo en la lista de candidatos. Se han mostrado anteriormente varios ejemplos para proporcionar el vector de reemplazo, que permiten procesamiento en paralelo de las unidades de procesamiento vecinas y/o aumentan la resiliencia de errores.

Ejemplo 2: Limitación de región

De acuerdo con un segundo ejemplo que no forma parte de la invención, que se puede combinar con el primer ejemplo y/o con cualquiera de los siguientes ejemplos, un bloque que está dentro de una región grande de $N \times M$ muestras no puede usar candidatos de vectores de movimiento refinados (o, en general, candidatos de vectores de movimiento determinados mediante al aplicar de coincidencia de plantillas) de un bloque vecino que también se encuentra dentro de la misma región. Por otro lado, el bloque puede usar candidatos de vectores de movimiento refinados de bloques vecinos o incluso vecinos que no se encuentran dentro de la misma región.

En otras palabras, la unidad de predicción puede configurarse para, para la reconstrucción del bloque actual, detectar si un vector de movimiento de un bloque circundante, que se encuentra dentro de una región predeterminada definida con respecto a la posición del bloque actual, se ha determinado por la coincidencia de plantillas. La región puede ser rectangular o incluso cuadrada. Sin embargo, la presente divulgación no se limita por lo mismo y la región puede tener cualquier otra forma.

Este enfoque se ilustra en la figura 8, paso S810, en el que se determina si el bloque vecino y el bloque actual están ubicados dentro de la misma ventana. Esta ventana puede ser la región de $N \times M$ muestras.

Las figuras 14 y 15 muestran ejemplos de las regiones de acuerdo con el segundo ejemplo que no forman parte de la invención.

En particular, la figura 14 muestra una región rectangular de $N \times M$ muestras. Los vectores de movimiento de bloques ubicados dentro de esta región no pueden usarse para derivar vectores de movimiento de bloques ubicados dentro de la misma región, especialmente si la derivación incluye la coincidencia de plantillas. Por otro lado, los vectores de movimiento de bloques ubicados fuera de la región, como el bloque B0, pueden usarse para derivar vectores de movimiento del bloque en la región, incluso si el bloque en la región y un bloque B0 son vecinos. Este enfoque permite procesamiento paralelo de los bloques que están dentro de la misma región. La región para un bloque actual puede definirse ventajosamente con respecto a la ubicación del bloque actual. En el ejemplo de la figura 14, la definición de "dentro de la región" significa que todo el bloque (actual y/o vecino), es decir, todas sus muestras, se encuentran en la región. Sin embargo, esta es solo una convención que no tiene ninguna consecuencia efectiva en el funcionamiento de este ejemplo.

La figura 15 muestra otro ejemplo, en el que la posición dentro de la región se determina solo por la ubicación de la muestra superior izquierda del bloque vecino o actual. La línea discontinua, como en el ejemplo anterior, muestra la región $N \times M$ muestras grandes (cuadrícula virtual T). Ahora, el bloque X, que es un bloque rectangular que se superpone con la región pero que no se encuentra completamente dentro de la región, también se considera que cae dentro de la región ya que su esquina superior izquierda se encuentra dentro de la región.

También se observa que, en general, la región no tiene que determinarse con respecto a la posición superior izquierda de un bloque, sino que puede determinarse con respecto a cualquier posición del bloque, tal como posición central.

En resumen, un bloque actual que está dentro de la región de $N \times M$ no puede usar vectores de movimiento de bloques ubicados en la misma región y procesados por derivación y/o refinamiento con base en la coincidencia de plantillas. En cambio, el bloque puede usar, por ejemplo, un vector de movimiento de reemplazo como se describe en el ejemplo 1 anterior. Por ejemplo, el vector de reemplazo es un vector de movimiento del bloque vecino dentro de la región antes del refinamiento, o es un vector de movimiento de un vecino (no adyacente al bloque actual) al bloque vecino, o es un vector de movimiento de otro bloque vecino.

El bloque actual dentro de la región de $N \times M$ puede usar vectores de movimiento de bloques ubicados fuera

de la región, independientemente de cómo se determinen. Esta regla se aplica independientemente de cómo se define el término dentro/fuera de la región (si todos los píxeles del bloque o solo el punto de referencia, tal como píxel superior izquierdo del bloque, tienen que estar dentro de la región).

5 Ejemplo 3: Omisión de las verificaciones

De acuerdo con un tercer ejemplo, el vector de movimiento de un bloque vecino determinado por la coincidencia de plantillas todavía se incluye en la lista de candidatos del bloque actual.

10 La construcción de una lista de candidatos tal como la lista de fusión o la lista AMVP puede incluir verificar la información de movimiento de múltiples bloques vecinos y agregarla a la lista de candidatos después de realizar la verificación de redundancia. La construcción de lista de fusión y los procesos de construcción de la lista AMVP (Advanced Motion Vector Predictor) (tal como se describe en el norma HEVC/H.265 y, por ejemplo, en el libro mencionado anteriormente por V. Sze et al, Sección 5.2.1 y 5.2.2) se refieren a la construcción de 2
15 listas de predictores de vectores de movimiento que se heredan de bloques vecinos temporales y espaciales. La selección de los bloques vecinos para heredar los vectores de movimiento es diferente (se emplean diferentes algoritmos) para las dos listas. Además, una vez que se selecciona el predictor de vector de movimiento, no se señala ninguna diferencia adicional del vector de movimiento en el flujo de bits, si se utiliza la lista de fusión. Por otro lado, la diferencia de MV se señala para el candidato AMVP.

20 La verificación de redundancia que se menciona en el párrafo anterior se refiere a identificación de información de movimiento que es la misma para 2 bloques vecinos. En otras palabras, durante los procesos de construcción de listas de fusión o AMVP, antes de agregar un vector de movimiento a la lista, se compara con los candidatos ya existentes (con todos ellos o con un subconjunto seleccionado) en la lista y se deja fuera de
25 la lista si ya hay un candidato idéntico en la lista (o en el subconjunto seleccionado). La operación de verificación de redundancia puede incluir comparación de 2 vectores de movimiento o información de movimiento completa (índice de imagen de referencia, modo de predicción (uni o bi-predicción y vectores de movimiento que apuntan a las imágenes de referencia que se utilizan).

30 Sin embargo, para permitir procesamiento paralelo de las unidades de procesamiento vecinas, no se verifica la redundancia de los vectores de movimiento refinados en la construcción de lista de candidatos de vectores de movimiento. Este enfoque tiene la ventaja de que la lista de candidatos (tal como la lista de fusión o lista AMVP) puede construirse sin conocer el valor real de los vectores de movimiento refinados o derivados. La reconstrucción real de vector de movimiento todavía necesita esperar el proceso de derivación y/o refinamiento.

35 En consecuencia, la unidad de determinación de lista de vectores de movimiento candidatos se configura además para verificar si un vector de movimiento que se incluirá en la lista ya está incluido en la lista, y no incluirlo en la lista si es afirmativo, y esta verificación no se realiza cuando se incluye en la lista un vector de movimiento de un bloque adyacente que se va a procesar al menos parcialmente en paralelo con el procesamiento del bloque actual.
40

Esto se ilustra en la figura 11. En la figura 11, se determina un vector de movimiento de la unidad de procesamiento PU1 que incluye derivación y refinamiento. Luego, el vector de movimiento refinado se ingresa en la lista de vectores de movimiento candidatos. En este ejemplo, se omite el proceso de verificación de
45 redundancia. Por lo tanto, la lista de candidatos de un bloque actual se puede construir antes de que se complete el proceso de refinamiento de vector de movimiento en un bloque vecino. Una vez que se completa el proceso de refinamiento, el vector de movimiento refinado de PU1 se incluye en la lista de candidatos pre-construidos de PU2. En otras palabras, la construcción de lista de candidatos no necesita esperar el vector de movimiento de PU1, sino puede completarse con los candidatos de vector de movimiento de otros bloques.
50 Luego se agrega el vector de movimiento de PU1 antes de realizar el paso de selección de vector de movimiento para PU2. Se pueden realizar en paralelo reconstrucción de textura y filtración de PU1 (bloque vecino) con el procesamiento de PU2 (bloque actual), en particular con la construcción de lista de candidatos, la selección/derivación y el refinamiento de vector de movimiento.

55 En el ejemplo anterior, la lista de fusión de PU2 se puede construir antes de que se complete la derivación de vector de movimiento o el refinamiento de PU1, ya que los valores reales de los vectores de movimiento de PU1 ya no son necesarios para el proceso de construcción de lista de fusión. Por lo tanto, PU1 puede continuar seleccionando el predictor candidato de la lista de fusión. En este paso, si el candidato predictor se hereda de PU1 (que aplica derivación o refinamiento de MV), PU1 y PU2 aún no se pueden procesar en paralelo, ya que los valores reales de los vectores de movimiento de PU1 no están disponibles y se requieren para la derivación
60 de MV/procesos de refinamiento y reconstrucción de textura. Sin embargo, si el candidato predictor se hereda de PU3 (que no aplica derivación/refinamiento de MV), PU1 y PU3 se pueden procesar en paralelo, ya que los valores reales de los vectores de movimiento de PU3 ya se conocen y están disponibles en este punto en el tiempo. Un implementador de decodificador puede facilitar el método inventado para acelerar el proceso de decodificación de una manera oportunista, es decir, empleando reconstrucción paralela de CU siempre que sea posible.
65

Ejemplo 4: Predicción limitada de vector de movimiento

En los ejemplos anteriores, los vectores de movimiento de bloques vecinos obtenidos por derivación y refinamiento de vector de movimiento aplicando coincidencia de plantillas no se usaron como vectores de movimiento candidatos para el bloque actual. De acuerdo con un cuarto ejemplo que no forma parte de la invención, el vector de movimiento de un bloque actual se predice usando el vector de movimiento de un bloque adyacente al bloque actual (preferiblemente directamente adyacente espacialmente, pero en general, también se pueden usar bloques temporalmente adyacentes). La diferencia con la predicción puede señalarse en el flujo de bits. Alternativamente, el proceso de refinamiento de vector de movimiento con base en decodificador puede realizarse en el vector de movimiento predicho.

La figura 12 ilustra un bloque actual con un bloque vecino izquierdo 1 cuyo vector de movimiento se ha determinado por derivación y/o refinamiento de vector de movimiento con base en la coincidencia de plantillas y con un vecino superior 2 del cual el vector de movimiento no se ha determinado por derivación de vector de movimiento y/o refinamiento con base en la coincidencia de plantillas. De acuerdo con el cuarto ejemplo, si el bloque actual aplica la coincidencia de plantillas para determinar el vector de movimiento, entonces no utiliza el vector de movimiento de un bloque vecino para la predicción de vector de movimiento si el vector de movimiento del bloque vecino se determinó con base en la coincidencia de plantillas. Por lo tanto, en la figura 12, el vector de movimiento del bloque vecino 1 no puede usarse para la predicción de vector de movimiento del vector de movimiento del bloque actual. Por otro lado, si el bloque actual no aplica la coincidencia de plantillas para su determinación del vector de movimiento, puede usar el vector de movimiento del bloque vecino 1, por ejemplo, para la predicción de vector de movimiento (es decir, calcular el vector de movimiento del bloque actual como la suma de una predicción determinada con base en el vector de movimiento del bloque vecino o más bloques vecinos y una diferencia señalada en el flujo de bits).

En cualquier caso, el vector de movimiento del bloque actual se puede predecir con base en el bloque vecino 2 si no se aplica la coincidencia de plantillas para determinar el vector de movimiento del bloque vecino 2. Se observa que la predicción de vector de movimiento para el bloque actual puede realizarse tomando como predicción directamente un vector de movimiento de un bloque vecino. Sin embargo, la predicción de vector de movimiento también puede realizarse con base en un vector de movimiento de un bloque vecino, por ejemplo promediando vectores de movimiento de varios bloques vecinos espaciales y/o temporales.

Ejemplo 5: Intra-predicción limitada

De acuerdo con un quinto ejemplo que no forma parte de la invención, la unidad de predicción comprende una unidad de determinación de muestra de referencia para determinar un conjunto de muestras de referencia para el bloque actual que incluye:

- detectar si un vector de movimiento de un bloque adyacente se ha determinado por la coincidencia de plantillas,
- si se detecta negativamente, incluir una o más muestras del bloque adyacente en el conjunto de muestras de referencia, y
- si se detecta positivamente, no incluir ninguna muestra del bloque adyacente en el conjunto de muestras de referencia,

Además, la unidad de predicción incluye una unidad de inter-predicción para proporcionar la predicción del bloque actual de acuerdo con muestras del conjunto de muestras de referencia.

Esto se ilustra en la figura 13, en la que el bloque actual está rodeado por un bloque vecino izquierdo 1 del cual el vector de movimiento se ha determinado aplicando coincidencia de plantillas y un bloque vecino superior 2 del cual el vector de movimiento no se ha determinado o no determinado mediante aplicación de coincidencia de plantillas. El bloque actual es un intra-bloque, es decir, su predicción se determinará mediante una inter-predicción basada en muestras de referencia (píxeles) de los bloques circundantes en la misma imagen. Dado que el bloque vecino 1 aplica derivación y/o refinamiento de vector de movimiento, la inter-predicción del bloque actual no usará muestras del bloque vecino 1. Dado que el bloque vecino 2 no aplica la derivación y/o refinamiento de vector de movimiento, la intra-predicción del bloque actual puede usar muestras reconstruidas del bloque vecino 2 para la interpredicción.

Como ya se mencionó anteriormente, incluso si las figuras 12 y 13 muestran solo un bloque vecino izquierdo y uno superior, en general puede haber más bloques vecinos como se muestra en la figura 9. En general, las muestras de referencia de bloques que usaron plantilla la coincidencia para derivar sus vectores de movimiento no se usan para inter-predicción del bloque actual.

Ejemplo 6: Señalización

Para permitir que el codificador y el decodificador funcionen de manera compatible, se puede introducir una señalización de control en el flujo de bits que especifica si el procesamiento en paralelo y/o la resiliencia se controlan y cómo se controlan como se describió anteriormente.

Como ya se describió con referencia a las figuras 1 y 2, un decodificador de video para decodificar una pluralidad de imágenes de un flujo de bits puede comprender un analizador de flujo de bits para analizar el flujo de bits que incluye información de control, la unidad de predicción como se describe en los ejemplos anteriores y una unidad de reconstrucción para reconstruir el bloque actual con base en la predicción del bloque actual.

Además, un codificador de video para codificar una pluralidad de imágenes en un flujo de bits puede comprender la unidad de predicción como se describe en los ejemplos anteriores, un formador de flujo de bits para inclusión en la información de control del flujo de bits y una unidad de reconstrucción para reconstruir el bloque actual con base en la predicción del bloque actual y almacenar el bloque reconstruido en una memoria.

El codificador o decodificador de video puede operar con la información de control que incluye: un indicador que indica si la unidad de predicción debe detectar si el vector de movimiento de un bloque adyacente se ha determinado por la coincidencia de plantillas y/o un indicador de reemplazo indicando, si se detecta positivamente, cómo se determina la predicción para el presente bloque.

En particular, el indicador de control puede indicar si se va a aplicar o no la restricción de acuerdo con la presente divulgación. Si el indicador indica que no se debe aplicar, no se realiza ninguna verificación y los vectores de movimiento de un bloque actual siempre se pueden derivar de los vectores de movimiento de los bloques vecinos, independientemente de cómo se hayan derivado los vectores de movimiento de los propios bloques vecinos. De lo contrario, se aplica una de las realizaciones y ejemplos anteriores. Se puede definir en una norma cuál de las realizaciones y ejemplos se aplica.

Además, es posible seleccionar la aplicación de una de las realizaciones y ejemplos descritos anteriormente, por ejemplo, se puede proporcionar una señalización de control que indique cómo se va a determinar el vector de reemplazo (vector de movimiento no refinado del vecino, vector de movimiento de un vecino del vecino o una copia de un vector de movimiento de otro vecino o cualquier otra cosa). Alternativamente o además, la señalización de control puede especificar si se va a aplicar uno o más del vector de reemplazo (ejemplo 1), limitación de región (ejemplo 2), omisión de verificaciones (ejemplo 3), predicción limitada de vector de movimiento (ejemplo 4) o intra-predicción limitada (ejemplo 5). También se puede indicar el tamaño de la región.

Por ejemplo, la información de control se incluye en un conjunto de parámetros de secuencia y/o en una cabecera de segmento.

Aquí, el término "conjunto de parámetros de secuencia" se refiere en general a cualquier campo de información de control que sea reconocido por el codificador y decodificador como aplicable a una pluralidad de imágenes que forman una secuencia. En particular, en el caso de HEVC, este puede ser el conjunto de parámetros de secuencia (SPS) o el conjunto de parámetros de imagen (PPS). Sin embargo, la presente divulgación no se limita a HEVC y, en general, se puede usar cualquier señalización común aplicable a más de una imagen. Esto puede ser información de control utilizada para todo el video o para un subconjunto de imágenes de video.

Además, el término "cabecera de segmento" en la presente se refiere a la información de control asociada con un segmento de video (imagen). HEVC utiliza segmentos de imágenes que pueden incluir la imagen completa o una parte de una imagen que pueden analizarse independientemente de otros segmentos. El análisis aquí incluye la decodificación entrópica. En algunos casos, el segmento también es decodificable de forma independiente, es decir, no utiliza información de otros segmentos para la decodificación mediante interpredicción. Sin embargo, en general, la información de control se puede proporcionar en asociación con cualquier porción de imagen a la que se proporciona señalización relacionada.

Se observa que la señalización de control puede aplicarse en ambos niveles. Por ejemplo, un indicador sobre si se debe usar o no la limitación en absoluto se puede señalar por segmento, mientras que la realización/ejemplo/tamaño de región particular se puede indicar por pluralidad de imágenes.

La determinación de vector de movimiento que incluye el refinamiento de vector de movimiento como se describió anteriormente puede implementarse como parte de la codificación y/o decodificación de una señal de video (imagen en movimiento). Sin embargo, la determinación de vector de movimiento también se puede usar para otros fines en el procesamiento de imágenes, tal como detección de movimiento, análisis de movimiento o similares.

Las figuras 16 y 17 ilustran escenarios en los que la presente invención, tal como se describió anteriormente,

- puede proporcionar ventajas relacionadas con una mejor resiliencia de errores. De acuerdo con la figura 16, se pierde un segmento que es necesario para reconstruir una porción de una imagen de referencia a la que apunta un bloque actual en un segmento actual. En estos casos, si los vecinos del bloque actual probablemente también apuntarán a la porción de segmento perdida. Si la porción de segmento está oculta, puede haber degradación de calidad de imagen. Si se retransmite, puede haber un mayor retraso de decodificador. De acuerdo con algunas realizaciones de ejemplo de la invención, los vectores de movimiento de vecinos que usan la imagen de referencia para determinar el vector de movimiento, no se usan para el bloque actual. En consecuencia, no habrá propagación de errores o retraso como resultado de la pérdida de corte.
- En la figura 17, no solo la imagen de referencia está dañada, sino también el vector de movimiento de los bloques vecinos que se utilizará en la lista de candidatos del bloque actual o que se utilizará para la predicción de vector de movimiento del bloque actual. Si se pierde un cuadro de referencia, las muestras de píxeles en el bloque actual no se pueden reconstruir. Además, los vectores de movimiento del bloque actual en el segmento actual no se pueden reconstruir en este ejemplo. Todos los vectores de movimiento en el sector actual y en los sectores anteriores están dañados. Dado que los siguientes bloques en orden de decodificación pueden usar los MV perdidos en la predicción de MV (fusión, AMVP, etc.), el decodificador no tiene una forma fácil de recuperación. El codificador no tiene medios para controlar propagación de errores (los bloques de codificación que preceden al bloque actual en el orden de codificación todos pueden estar dañados).
- De acuerdo con la invención descrita anteriormente, si se pierde una de las imágenes de referencia en la figura 17, aunque los píxeles de bloque actual están dañados (debido al hecho de que se pierden las muestras de referencia), se garantiza que los vectores de movimiento que utiliza el bloque están garantizados se reconstruyan correctamente. Ser capaz de reconstruir los vectores de movimiento correctamente (que es el beneficio proporcionado por los métodos inventados) proporciona los siguientes 2 beneficios:
- El decodificador puede emplear algoritmos de ocultación de errores. Dado que las imágenes consecutivas en una secuencia de video están altamente correlacionadas, un decodificador inteligente simplemente puede reemplazar una imagen perdida con una imagen de referencia disponible, modificar en escala los vectores de movimiento en consecuencia y obtener una predicción bastante buena para el bloque actual. Este método simple de ocultación de errores es posible debido al hecho de que el vector de movimiento del bloque actual pudo reconstruirse correctamente.
 - Dado que los siguientes bloques de codificación (dentro de la vecindad espacial o temporal) pueden usar el vector de movimiento del bloque actual como predictor de vector de movimiento, poder reconstruir el vector de movimiento del bloque actual garantiza que los vectores de movimiento de los siguientes bloques de codificación también se reconstruyen correctamente. Como resultado, un siguiente bloque de codificación que tiene todas sus imágenes de referencia disponibles (no perdidas) ahora se puede reconstruir perfectamente ya que su imagen de referencia y sus vectores de movimiento son correctos. Si no se aplica la invención, una imagen de referencia perdida hace que tanto los bloques de codificación actuales como los siguientes tengan vectores de movimiento incorrectos y el error se propaga a través de toda la secuencia de video (como resultado, solo un cuadro perdido puede causar que toda la secuencia de video se dañe).
- Resumiendo, la presente divulgación se refiere a inter-predicción en la que se determina si un vector de movimiento de un bloque vecino al bloque actual se ha determinado por la coincidencia de plantillas (como se usa, por ejemplo, en derivación y/o refinamiento de vector de movimiento basado en decodificador). Si se detecta negativamente, la predicción del bloque actual se deriva de acuerdo con el vector de movimiento del bloque vecino. Por otro lado, si se detecta positivamente, la predicción del bloque actual no se basa en el bloque vecino.
- La determinación del vector de movimiento puede implementarse como un aparato. Este aparato puede ser una combinación de un software y hardware. Por ejemplo, la determinación del vector de movimiento puede realizarse mediante un chip tal como un procesador de propósito general, o un procesador de señales digitales (DSP), o un arreglo de compuertas programables en el campo (FPGA), o similar. Sin embargo, la presente invención no se limita a la implementación en un hardware programable. Se puede implementar en un circuito integrado de aplicación específica (ASIC), o mediante una combinación de software y uno o más de los componentes de hardware mencionados anteriormente. En un ejemplo, el aparato comprende un procesador y una memoria que porta un programa, y el programa, cuando se ejecuta por el procesador, instruye al procesador a llevar a cabo los pasos de los métodos descritos anteriormente.
- La determinación del vector de movimiento también puede implementarse mediante instrucciones de programa almacenadas en un medio legible por computadora. El programa, cuando se ejecuta, hace que la computadora realice los pasos de detectar si un vector de movimiento de un bloque adyacente se ha determinado por la coincidencia de plantillas, si se detecta negativamente, derivar predicción del bloque actual de acuerdo con el bloque adyacente, y si se detecta positivamente, no derivar predicción del bloque actual de acuerdo con el bloque adyacente. El medio legible por computadora puede ser cualquier medio en el que se almacene el programa, tal como un DVD, CD, unidad USB (flash), disco duro, almacenamiento en el servidor disponible a

través de una red, etc.

5 El codificador y/o decodificador puede implementarse en varios dispositivos, que incluye un televisor, decodificador, PC, tableta, teléfono inteligente o similares. Puede ser una aplicación de software que implementa los pasos de método.

10 Aunque la invención se ha descrito anteriormente principalmente en el marco de codificación de video de imágenes en movimiento, las técnicas propuestas también se pueden aplicar para codificar (es decir, codificar o decodificar) cualquier conjunto de imágenes que comprenda dos o más imágenes. El conjunto de imágenes puede comprender, por ejemplo, un conjunto de imágenes fijas obtenidas en una aplicación imagenología, por ejemplo, una secuencia de imágenes de escaneo de tomografía computarizada (TC). Por lo tanto, en las reivindicaciones adjuntas, el término "video" puede significar una secuencia de imágenes en movimiento o cualquier otro conjunto de imágenes que comprenda dos o más imágenes.

REIVINDICACIONES

1. Un decodificador para determinar una predicción de un bloque actual basándose en una predicción de vector de movimiento del bloque actual, estando el decodificador configurado para:
 - a.) recibir señalización de control de un codificador, especificando la señalización de control si el decodificador debe aplicar un primer modo para determinar la predicción de vector de movimiento del bloque actual o si el decodificador debe aplicar un segundo modo para determinar la predicción de vector de movimiento del bloque actual;
 - b.) determinar, basándose en la señalización de control, si el decodificador debe aplicar el primer modo o el segundo modo para determinar la predicción de vector de movimiento del bloque actual;
 - c.) en caso de que se determine en el paso b.) que el decodificador debe aplicar el primer modo para determinar la predicción de vector de movimiento del bloque actual:
 - c1.) detectar, para cada uno de los múltiples bloques adyacentes al bloque actual, si se ha determinado o no un vector de movimiento del bloque adyacente al bloque actual por coincidencia de plantillas,
 - o si el vector de movimiento del bloque adyacente al bloque actual no se ha determinado por coincidencia de plantillas, incluir el vector de movimiento del bloque adyacente como un vector de movimiento candidato para la predicción de vector de movimiento del bloque actual en una primera lista,
 - o si el vector de movimiento del bloque adyacente al bloque actual se ha determinado por coincidencia de plantillas, no incluir el vector de movimiento del bloque adyacente en la primera lista, sino incluir un vector de movimiento de reemplazo como un vector de movimiento candidato para la predicción de vector de movimiento del bloque actual en la primera lista,
 - c2.) seleccionar un primer vector de movimiento de la primera lista como una estimación de la predicción de vector de movimiento del bloque actual, determinar la predicción de vector de movimiento del bloque actual de acuerdo con la estimación y realizar la predicción del bloque actual basándose en la predicción del vector de movimiento seleccionado del bloque actual;
 - d.) en caso de que se determine en el paso b.) que el decodificador debe aplicar el segundo modo para determinar la predicción de vector de movimiento del bloque actual:
 - d1.) detectar, para cada uno de los múltiples bloques adyacentes al bloque actual, si se ha determinado o no el vector de movimiento del bloque adyacente al bloque actual por coincidencia de plantillas,
 - d11.) si el vector de movimiento del bloque adyacente al bloque actual no se ha determinado por coincidencia de plantillas, verificar si el vector de movimiento del bloque adyacente al bloque actual ya está incluido en una segunda lista,
 - o si el vector de movimiento del bloque adyacente al bloque actual ya está incluido en la segunda lista, no incluir el vector de movimiento del bloque adyacente al bloque actual en la segunda lista;
 - o si el vector de movimiento del bloque adyacente al bloque actual no está incluido en la segunda lista, incluir el vector de movimiento del bloque adyacente al bloque actual en la segunda lista;
 - d12.) si el vector de movimiento del bloque adyacente al bloque actual se ha determinado por coincidencia de plantillas, no realizar la verificación, pero agregar el vector de movimiento del bloque adyacente al bloque actual en la segunda lista;
 - d2.) seleccionar un segundo vector de movimiento de la segunda lista como una estimación de la predicción de vector de movimiento del bloque actual, determinar la predicción de vector de movimiento del bloque actual de acuerdo con la estimación, y realizar la predicción del bloque actual basándose en la predicción de vector de movimiento seleccionado del bloque actual.
2. El decodificador de acuerdo con la reivindicación 1, en donde el vector de movimiento de reemplazo es un vector de movimiento de un bloque vecino al bloque actual y diferente del bloque adyacente.
3. Un método para determinar una predicción de un bloque actual basándose en una predicción de vector de movimiento del bloque actual realizada por un decodificador, el método comprende los pasos de:
 - a.) recibir señalización de control de un codificador, especificando la señalización de control si el decodificador debe aplicar un primer modo para determinar la predicción de vector de movimiento del bloque actual o si el

decodificador debe aplicar un segundo modo para determinar la predicción de vector de movimiento del bloque actual;

5 **b.)** determinar, basándose en la señalización de control, si el decodificador debe aplicar el primer modo o el segundo modo para determinar la predicción de vector de movimiento del bloque actual;

c.) en caso de que se determine en el paso **b.)** que el decodificador debe aplicar el primer modo para determinar la predicción de vector de movimiento del bloque actual:

10 **c1.)** detectar, para cada uno de los múltiples bloques adyacentes al bloque actual, si se ha determinado o no un vector de movimiento del bloque adyacente al bloque actual por coincidencia de plantillas,

15 ◦ si el vector de movimiento del bloque adyacente al bloque actual no se ha determinado por coincidencia de plantillas, incluir el vector de movimiento del bloque adyacente como un vector de movimiento candidato para la predicción de vector de movimiento del bloque actual en una primera lista,

20 ◦ si el vector de movimiento del bloque adyacente al bloque actual se ha determinado por coincidencia de plantillas, no incluir el vector de movimiento del bloque adyacente en la primera lista, sino incluir un vector de movimiento de reemplazo como un vector de movimiento candidato para la predicción de vector de movimiento del bloque actual en la primera lista,

25 **c2.)** seleccionar un primer vector de movimiento de la primera lista como una estimación de la predicción de vector de movimiento del bloque actual, y determinar la predicción de vector de movimiento del bloque actual de acuerdo con la estimación y realizar la predicción del bloque actual basándose en la predicción de vector de movimiento seleccionado del bloque actual;

d.) en caso de que se determine en el paso **b.)** que el decodificador debe aplicar el segundo modo para determinar la predicción de vector de movimiento del bloque actual:

30 **d1.)** detectar, para cada uno de los múltiples bloques adyacentes al bloque actual, si se ha determinado o no el vector de movimiento del bloque adyacente al bloque actual por coincidencia de plantillas,

35 **d11.)** si el vector de movimiento del bloque adyacente al bloque actual no se ha determinado por coincidencia de plantillas, verificar si el vector de movimiento del bloque adyacente al bloque actual ya está incluido en una segunda lista,

 ◦ si el vector de movimiento del bloque adyacente al bloque actual ya está incluido en la segunda lista, no incluir el vector de movimiento del bloque adyacente al bloque actual en la segunda lista;

40 ◦ si el vector de movimiento del bloque adyacente al bloque actual no está incluido en la segunda lista, incluir el vector de movimiento del bloque adyacente al bloque actual en la segunda lista;

45 **d12.)** si el vector de movimiento del bloque adyacente al bloque actual se ha determinado por coincidencia de plantillas, no realizar la verificación, sino agregar el vector de movimiento del bloque adyacente al bloque actual a la segunda lista;

50 **d2.)** seleccionar un segundo vector de movimiento de la segunda lista como una estimación de la predicción de vector de movimiento del bloque actual, determinar la predicción de vector de movimiento del bloque actual de acuerdo con la estimación, y realizar la predicción del bloque actual basándose en la predicción de vector de movimiento seleccionado del bloque actual.

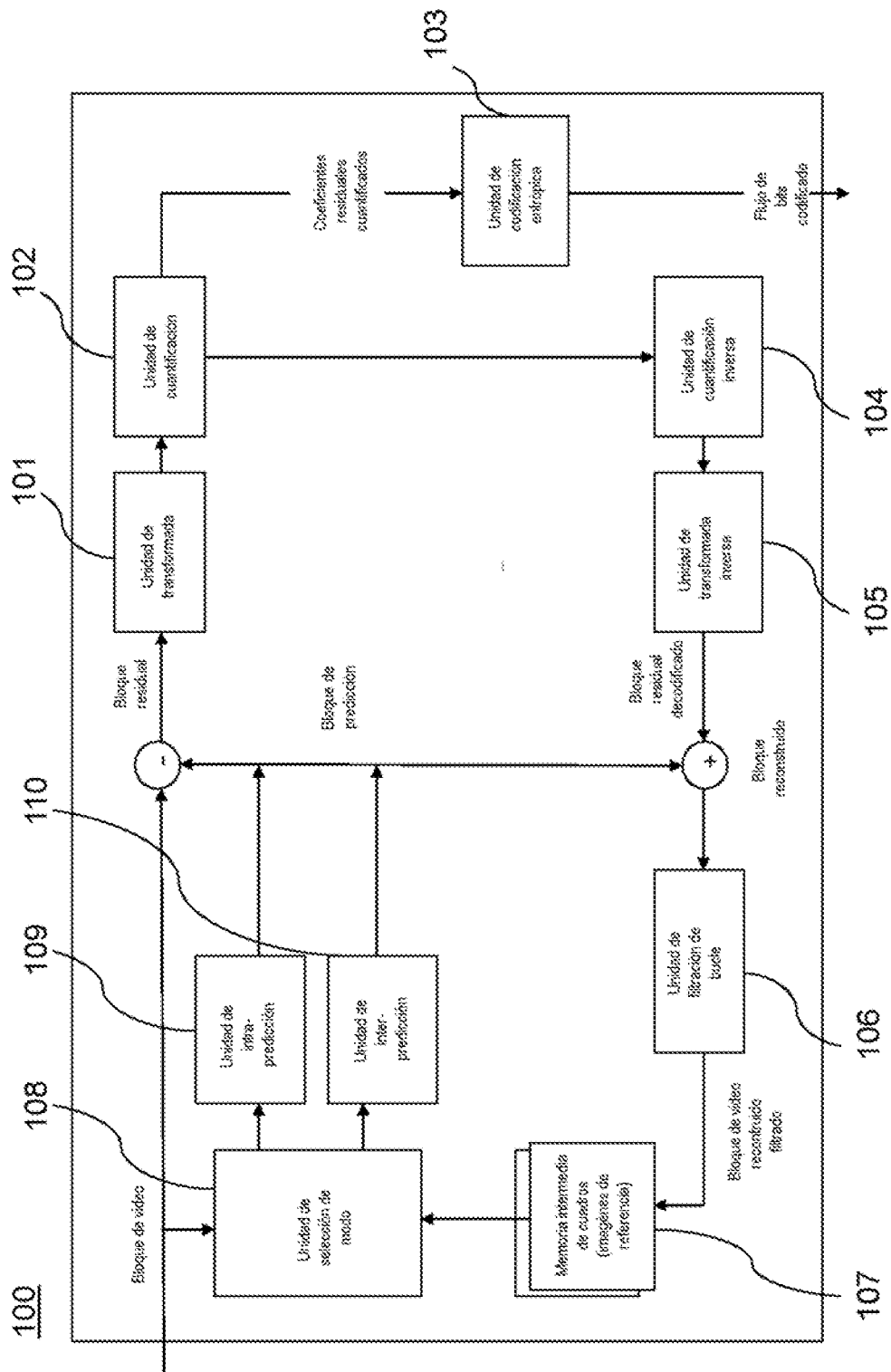


Fig. 1

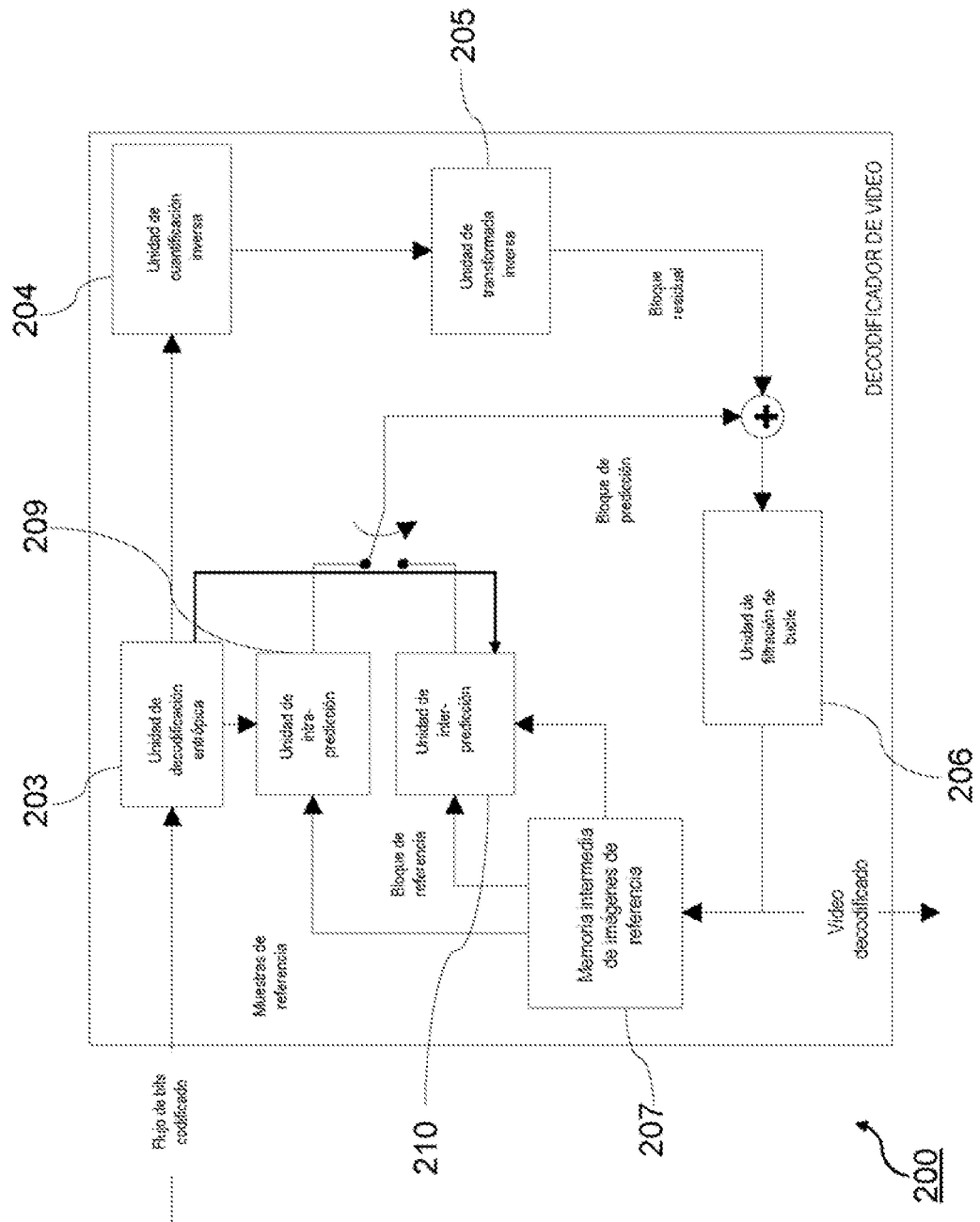


Fig. 2

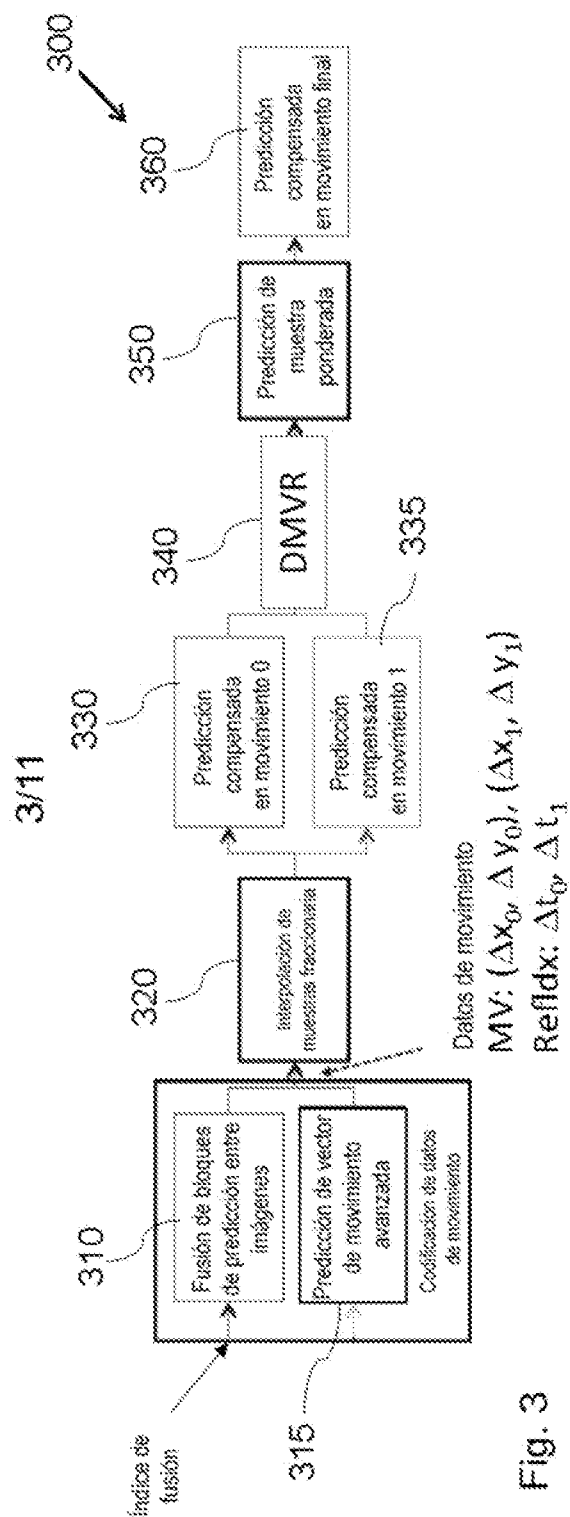


Fig. 3

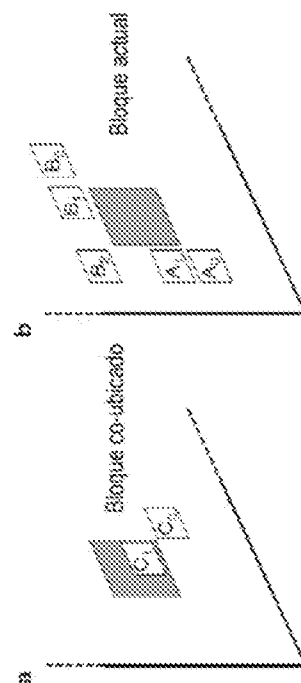


Fig. 4

4/11

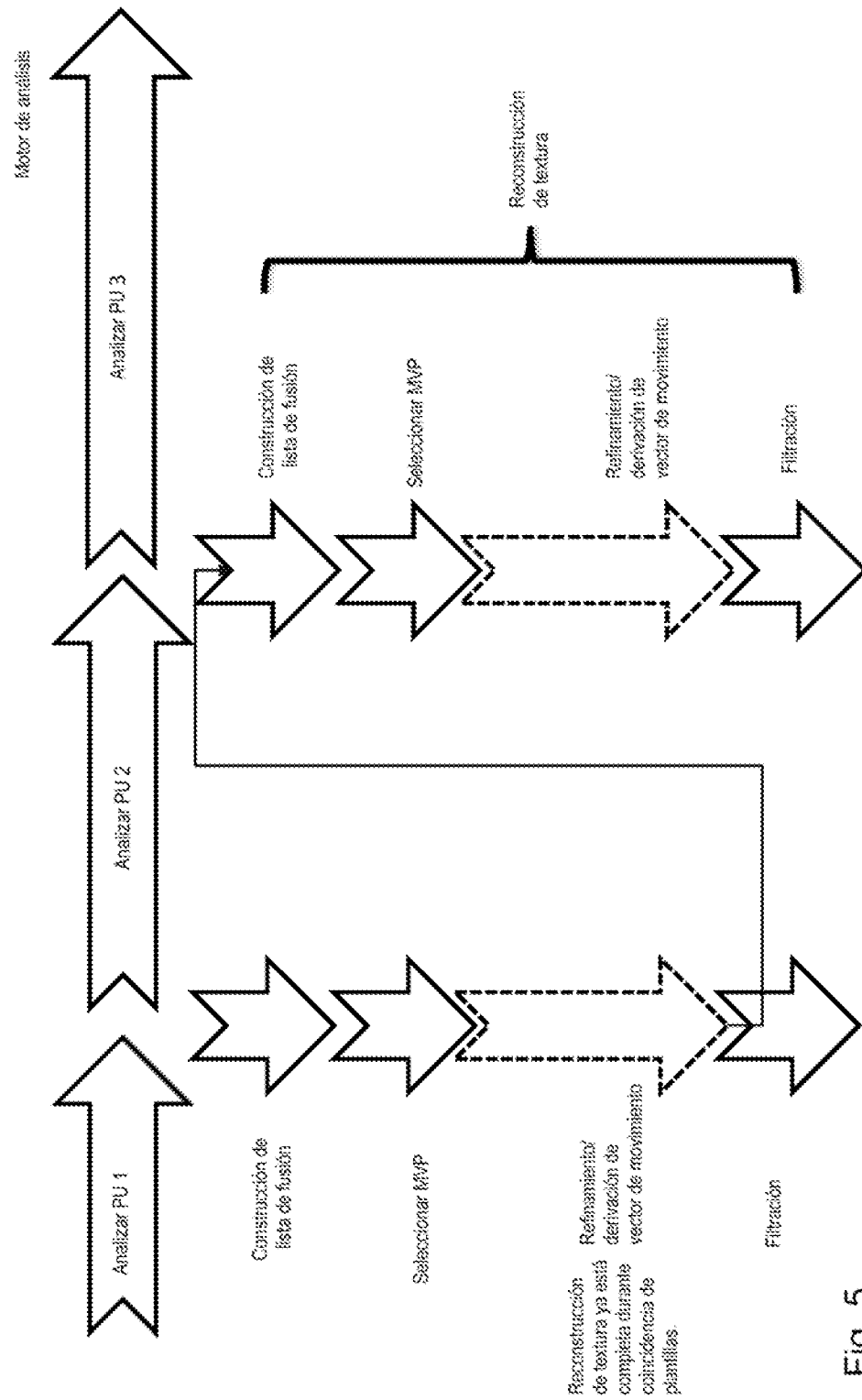


Fig. 5

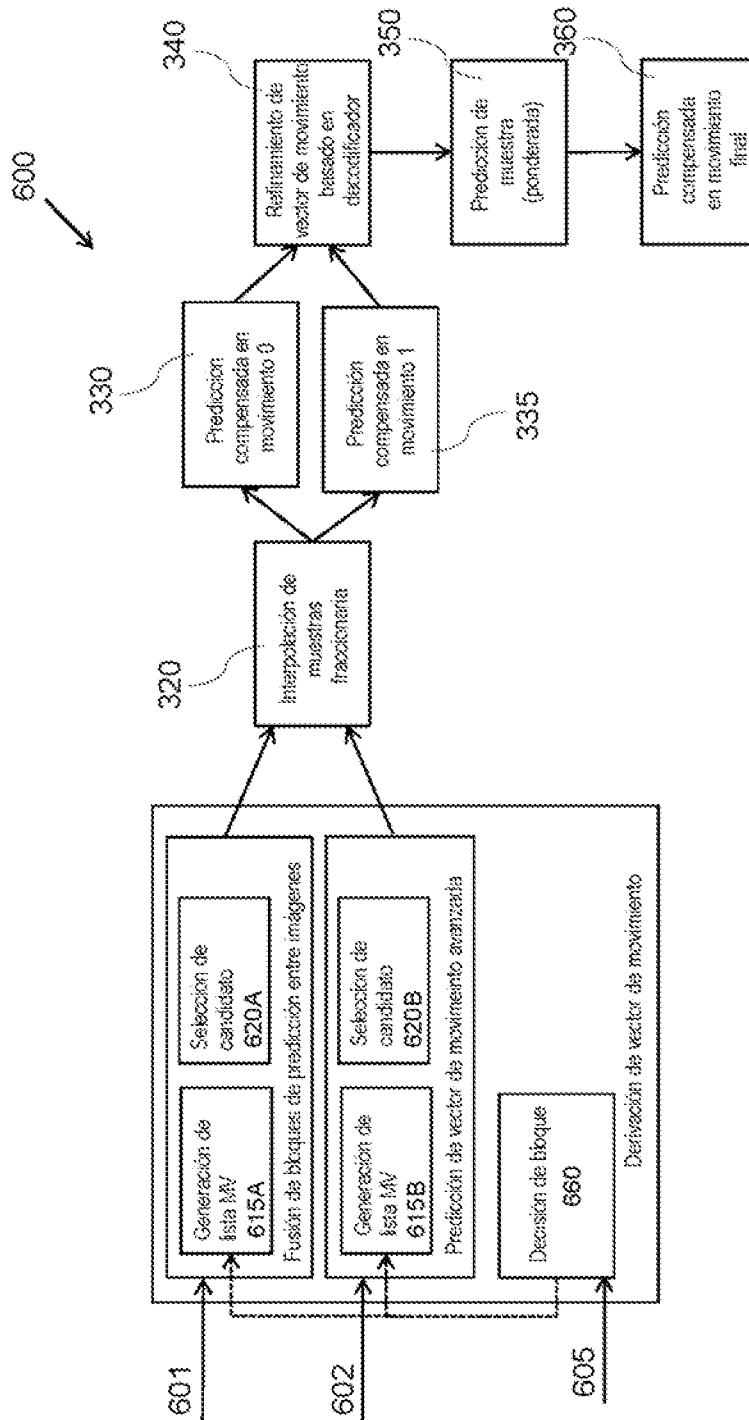


Fig. 6

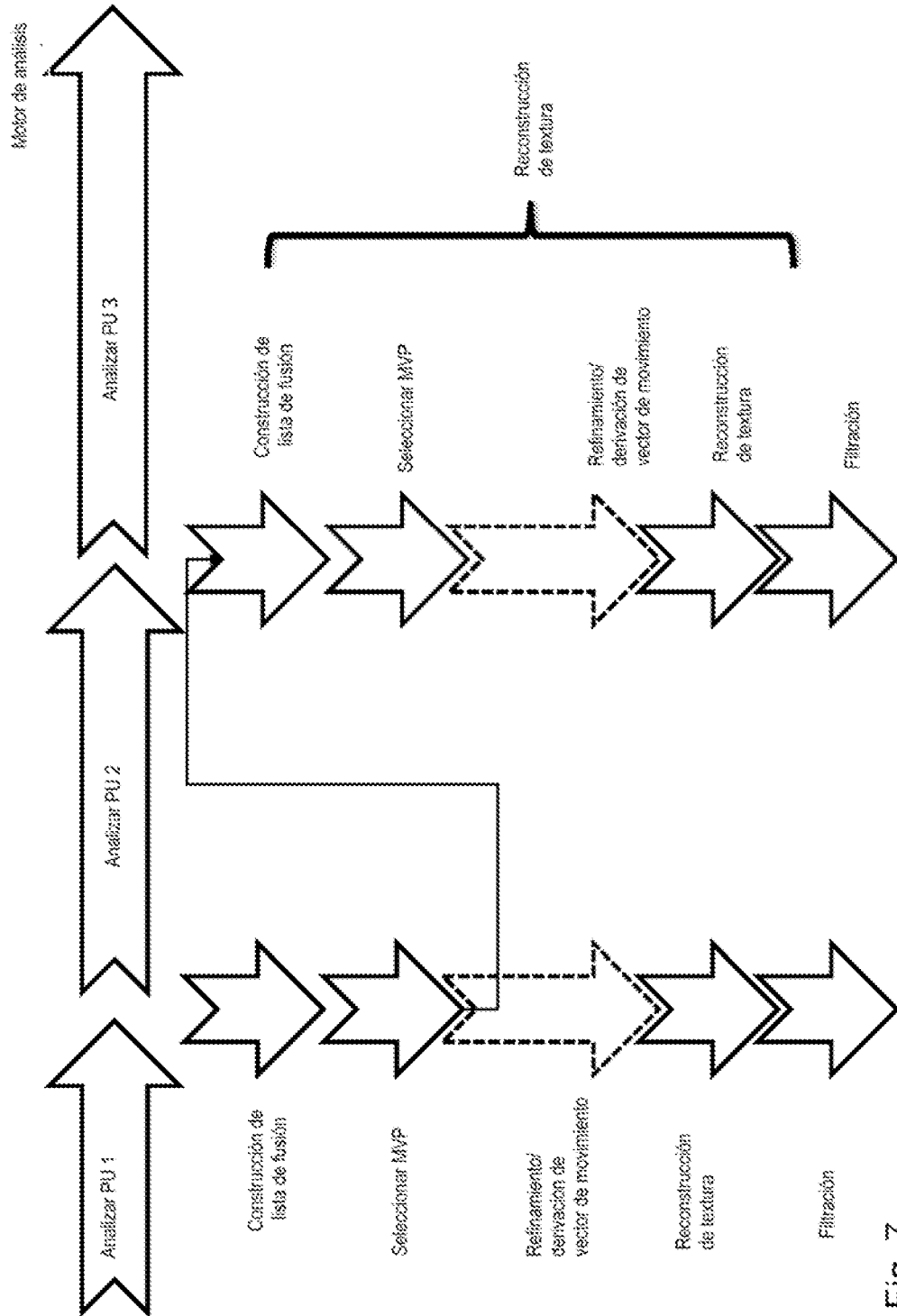


Fig. 7

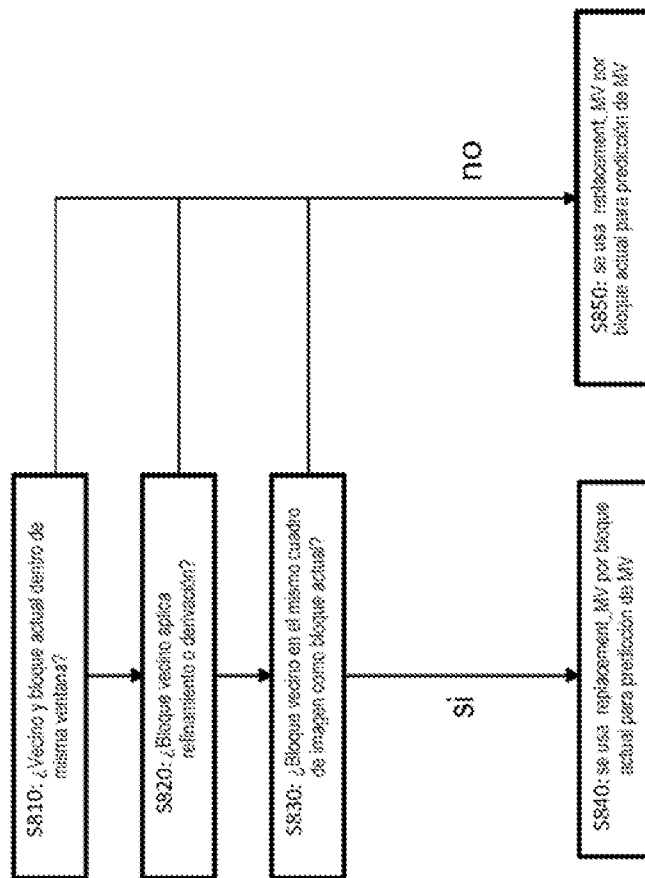


Fig. 8

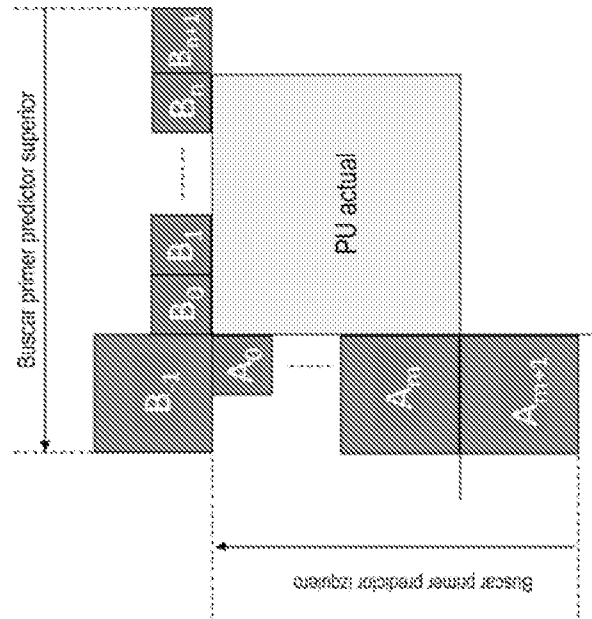


Fig. 9

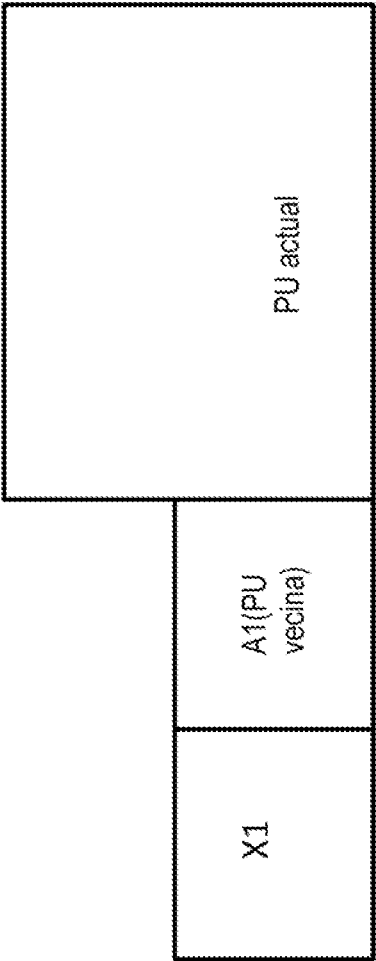


Fig. 10

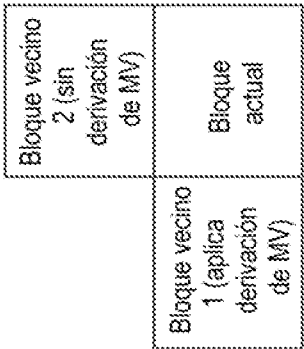


Fig. 12

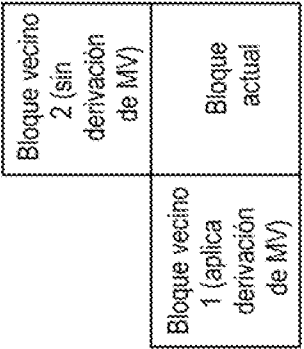


Fig. 13

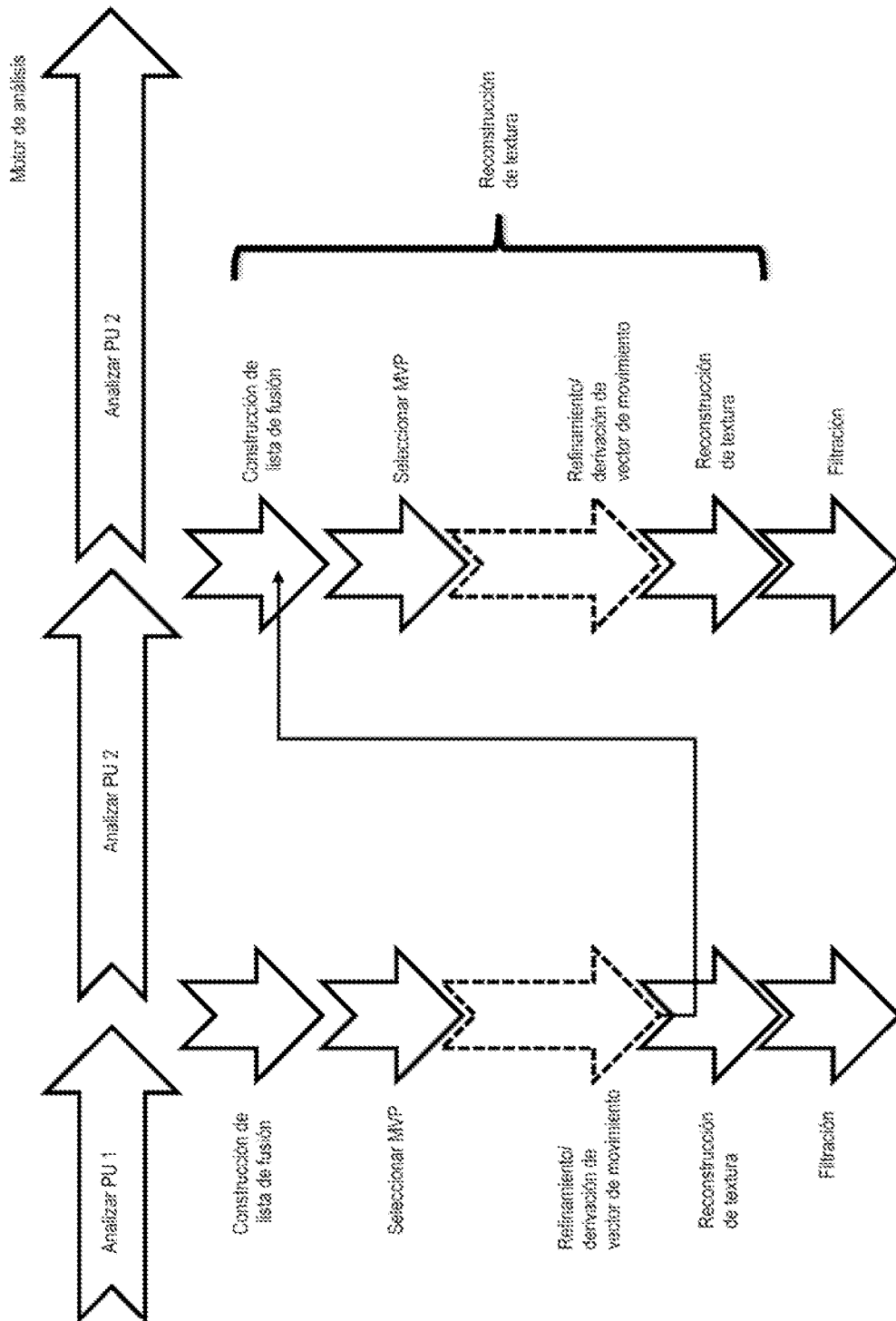


Fig. 11

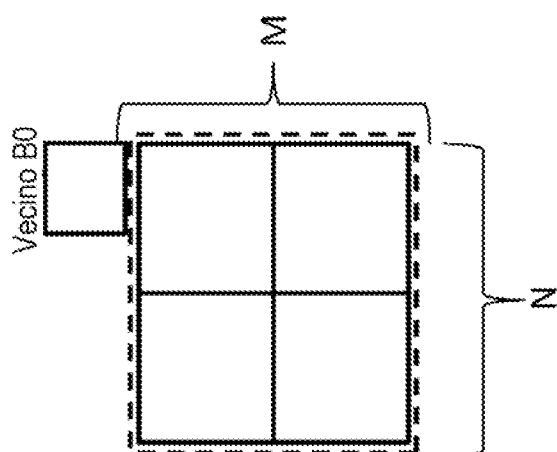


Fig. 14

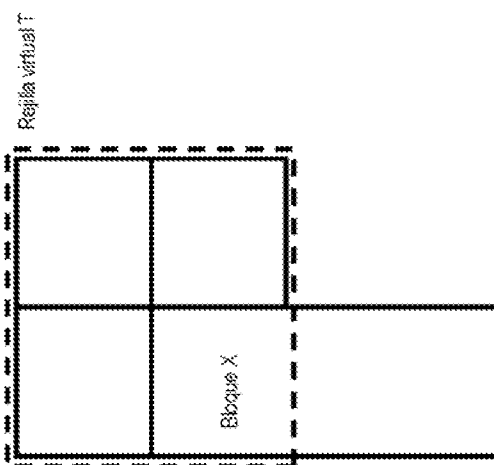


Fig. 15

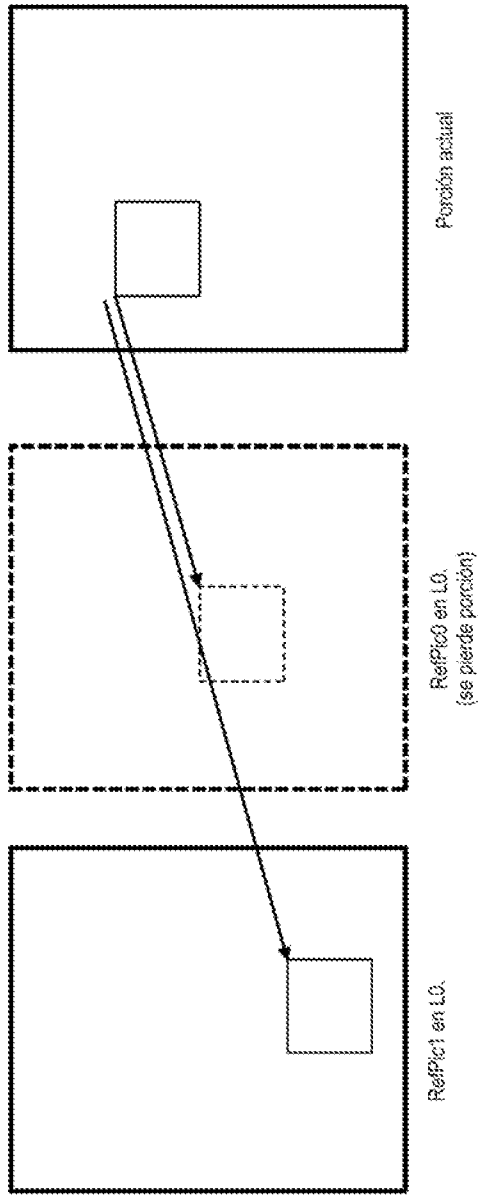


Fig. 16

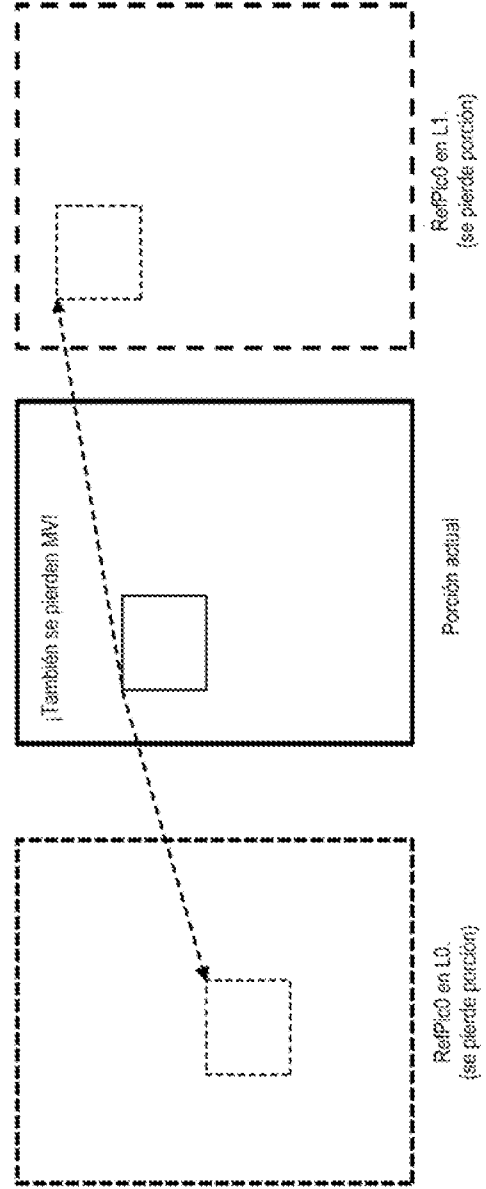


Fig. 17